

**MODELE DE REZOLVARE
A SUBIECTELOR DE BACALAUREAT 2008**

**INFORMATICĂ
intensiv și neintensiv**

material descărcat de pe site-ul eBacalaureat.ro

Descrierea CIP a Bibliotecii Naționale a României

**Metode de rezolvare a subiectelor de bacalaureat 2008 :
informatică intensiv și neintensiv/ coord.: Giorgie
Daniel Vlad, Ovidiu Marian Marcu. - Suceava : Little
Lamb, 2008**

ISBN 978-973-7762-63-4

I. Vlad, Giorgie Daniel (coord.)

II. Marcu, Ovidiu (coord.)

004

**MODELE DE REZOLVARE
A SUBIECTELOR DE BACALAUREAT 2008**

INFORMATICĂ

intensiv și neintensiv

*în conformitate cu modificările prevăzute
în erata publicată pe situl MECT,
în luna aprilie, 2008*

Autori:

Numele și prenumele	Județ	Variante
Marinescu Minodora	Arad	1 – 7
Vătămănescu Nicu Petrișor Adrian Costineanu Raluca Hatmanu Mihaela	Suceava	10, 11 8, 9 14 12, 13
TarasăDaniela	Bac ău	15 – 21
Mureșan Ciprian	Bistrița Năsăud	22 – 28
Voinea-Axinte Alina	Botoșani	29 – 35
Prus Paul	Brăila	36 – 42
Istrate Nicolae Cecilian , Vitan Adrian	Dâmbovița	43 - 49
Bradatan Claudița (52) Popa Carmen (51) Pintea Eugen (53) Ududec Marius si Stefanescu Narcisa (54) Erhan Mihail (56) Zotic Cristina (50) Aga Petronela (55)	Suceava	52 51 53 54 56 50 55
Balacea Georgeta Iulia	Galați	57 – 63
Runceanu Mihaela	Gorj	64 - 70
Jacobescu Raluca Silvia Hadimbu Stelian	Hunedoara Iași	71 – 77 78 – 84
Ungureanu Florentina	Neamț	85 – 91
Vlad Giorgie Daniel (coordonator) Marcu Ovidiu Marian (coordonator) Marcu Daniela HojbotăRodica	Suceava	92 – 93 94 – 97 98 – 100 Varianta Pascal: 13 (II. 49) 63 (III. 3)

Coperta și tehnoredactare: **Daniela Marcu**

SUBIECTUL I

1. d

2.

b) 89, 65, 67

c)

citește n (număr natural)

z ← 0

p ← 1

dacă n <> 0 atunci

repetă

c ← n % 10

n ← [n / 10]

dacă c % 3 = 0 atunci

z ← z + p * (9 - c)

p ← p * 10

până când n = 0

d) Varianta C/C++

#include <iostream.h>

void main()

{

unsigned int n, c, z = 0, p = 1;

cout << "n="; cin >> n;

while (n > 0)

{

c = n % 10;

n = n / 10;

if (c % 3 == 0)

{

z = z + p * (9 - c);

p = p * 10;

}

}

cout << z;

}

Varianta PASCAL

var n, c, z, p: word;

begin

z := 0; p := 1;

write('n='); readln(n);

while n > 0 do

begin

c := n mod 10;

n := n div 10;

if c mod 3 = 0 then

begin

z := z + p * (9 - c);

p := p * 10;

end

end;

write(z)

end.

SUBIECTUL II

1. c

2. d

3.

Varianta C/C++

C.x = (A.x + B.x) / 2;

C.y = (A.y + B.y) / 2;

4.

Pentru intensiv

Înălțimea arborelui este 3

Frunzele arborelui sunt: 1, 2, 3 și 8

Pentru neintensiv

Nodul cu cei mai mulți fii este 6.

Varianta PASCAL

C.x := (A.x + B.x) / 2;

C.y := (A.y + B.y) / 2;

Frunzele arborelui sunt: 1, 2, 3 și 8

5.

Varianta C/C++

```
#include<iostream.h>
void main()
{
    unsigned int
n,m,i,j,A[11][11];
    do
    {cout<<"n=";cin>>n;
    }
    while(n<2 || n>10);
    do
    {cout << "m"; cin>> m;
    }
    while ( m<2 || m>10);
    for (i=1;i<=n;i++)
        for (j=1;j<=m;j++)
            if(i<j) A[i][j]=i;
            else A[i][j]=j;
    for (i=1;i<=n;i++)
    {
        for (j=1;j<=m;j++)
            cout <<A[i][j]<<` `;
        cout <<`\n`;
    }
}
```

Varianta PASCAL

```
var n,m,,i,j:byte;
    A:array[1..10,1..10] of byte;
begin
    repeat
        write(`n=`);readln(n);
    until (n>=2) and (n<=10);
    repeat
        write(`m=`);readln(m);
    until (m>=2) and (m<=10);
    for i:=1 to n do
        for j:=1 to m do
            if i<j then A[i,j]:=i
            else A[i,j]:=j;
    for i:=1 to n do
        begin
            for j:=1 to m do
                write(A[i,j],` `);
            writeln;
        end
    end.
```

ȘUBIECTUL III

1. b

2. 12345 pentru **intensiv**

6 pentru **neintensiv** în cazul în care prima condiție a funcției în C/C++ este n==0

3.

Varianta C/C++

Pentru intensiv

```
# include "stdio.h"
void main()
{
    unsigned n,a,este=0;

    FILE *f;"bac.txt","r");
    do
    { printf("n=")
    scanf("%d", &n);
    }
    while (n==0 || n>999);
    while(!feof(f))
    {
        fscanf(f,"%d",&a);
        if(a%n==0)
            {printf("%d ",a);
            este=1;
            }
    }
    if(este==0)
        printf("NU
```

Varianta PASCAL

```
var f:text;
    n,a:word;
    este:boolean;
begin
    assign(f,'bac.txt');
    reset(f);
    repeat
        write(`n=`);readln(n);
    until (n<>0) and (n<=999);
    este:=false;
    while not(eof(f)) do
        begin
            read(f,a);
            if a mod n=0 then
                begin
                    write(a,` `);
                    este:=true;
                end;
        end;
    if not este then
        write(`NU EXISTA');
```

```
EXISTA");
    fclose(f);
}
```

4. a)

Varianta C/C++

Pentru intensiv

```
int sub (int v[100],int n,int
a)
{int i; nr=0 ;
for(i=0; i<n; i++)
    if (v[i]<a) nr++;
return nr;
}
```

La neintensiv se modifică doar condiția v[i]=a în loc de v[i]<a

b)

Varianta C/C++

Pentru intensiv

```
#include <iostream.h>
#include <conio.h>
int v[100],int n, int a;
{
int i, nr=0 ;
for(i=0;i<n;i++)
return if(v[i]<a) nr++ ;
}

void main()
{
int i,j,este=1;
do
{
cout<<"n= "; cin>>n ;
}
while(n==0 || n>100);
for(i=0;i<n;i++)
do
{
cout <<"v["<<i<<"]=";
cin>>v[i] ;
}
while(v[i]<-9999 || v[i]>9999);
for(i=0;i<n;i++)
t[i]=sub(v,n,v[i]);
i=0;
while(i<n && este)
{
este=0;
for(j=0;j<n;j++)
if (j==t[j] && j==i) este=1;
i++;
}
```

```
close(f);
end.
```

Varianta PASCAL

```
type vector=array[1..100] of integer;
...
function
sub(v:vector;n:integer;a:integer):integer;
var i,nr:integer;
begin
nr:=0;
for i:=1 to n do
if v[i]<a then nr:=nr+1;
sub:=nr;
end;
```

La neintensiv se modifică doar condiția v[i]=a în loc de v[i]<a

Varianta PASCAL

```
type vector=array[1..100] of integer;
var v:vector;
n,a,i,j:integer;
este:boolean;
function
sub(v:vector;n:integer;a:integer):integer;
var i,nr:integer;
begin
nr:=0;
for i:=1 to n do
if v[i]<a then nr:=nr+1;
sub:=nr;
end;
begin
este:=true;
repeat
write('n=');readln(n);
until (n<>0) and (n<=100);
for i:=1 to n do
repeat
write('v['<i',']=');
readln(v[i]);
until (v[i]>=-9999) and (v[i]<=9999);
for i:=1 to n do
t[i]:=sub(v,n,v[i]);
i:=0;
while (i<n) and este=true do
begin
este:=0;
for j:=1 to n do
if i=t[j] then este:=true;
i:=i+1;
end;
if este=true then write('DA')
else write('NU');
```

```
if(este) cout<<"DA";
else cout <<"NU";
}
```

Pentru neintensiv

```
#include <iostream.h>
#include <conio.h>
int v[100],int n, int a;
{
    int i, nr=0 ;
    for(i=0;i<n;i++)
        if(v[i]==a) nr++; ;
return nr;
}

void main()
{
int i;

do
{
    cout<<"n= "; cin>>n ;
}
while(n==0 || n>100);
for(i=0;i<n;i++)
do
{
    cout <<"v["<<i<<"]="";
    cin>>v[i] ;
}
while(v[i]<-9999 || v[i]>9999);
for(i=0;i<n;i++)
    t[i]=sub(v,n,v[i]);
i=0;
while(i<n && t[i]==1) i++;
if(i<n) cout<<"NU";
else cout <<"DA";
}
```

end.

```
type vector=array[1..100] of integer;
var v:vector;
    n,a,i:integer;
function
sub(v:vector;n:integer;a:integer):integer;
var i,nr:integer;
begin
    nr:=0;
    for i:=1 to n do
        if v[i]=a then nr:=nr+1;
    sub:=nr;
end;
begin
    repeat
        write('n=');readln(n);
    until (n<>0) and (n<=100);
    for i:=1 to n do
        repeat
            write('v[',i,']=');
            readln(v[i]);
        until (v[i]>=-9999)and(v[i]<=9999);
    for i:=1 to n do
        t[i]:=sub(v,n,v[i]);
    i:=0;
    while (i<n) and (t[i]=1) do i:=i+1;
    if i<n then write('NU')
    else write('DA');
end.
```

Varianta <2>

SUBIECTUL I

1. a

2.

a) 2211775

b) 9 8 7 7 0

c)

citește x (număr natural nenul)

```
repetă
┌ citește y (număr natural)
│ ┌ dacă x>y atunci
│ │ scrie x%10
│ │ altfel
│ │ scrie y%10
└ ┌ x←y
  └ până când y=0
```


d) Varianta C/C++

```
#include<iostream.h>
void main()
{
    unsigned int x,y;
    do
    {
        cout<<"x=";>>x;
    }
    while(x==0);
    while(x>0)
    {
        cout<<"y=";>>y;
        if(x>y) cout<<x%10;
        else cout<<y%10;
        x=y;
    }
}
```

Varianta PASCAL

```
var x,y:word;
begin
    repeat
        write('x=');readln(x);
    until x>0;
    while x>0 do
        begin
            write('y=');
            readln(y);
            if x>y then write(x mod 10)
            else write(y mod 10);
            x:=y;
        end;
    end.
```

SUBIECTUL II

1. a

2. c

3. Valorile ultimelor douăelemente eliminate sunt 5 și 2.

4. Varianta C/C++

Pentru intensiv

```
s:=0;
for (j=0;j<n/2;j++)
{
    if (a[k][2*j]%2==1)
        s=s+a[k][2*j];
    if (a[k][2*j+1]%2==1)
        s=s+a[k][2*j+1];
}
printf("%d",s)
```

Pentru neintensiv

```
s:=0;
for (i=0;i<n;i++)
    if (a[k][i]%2==1)
        s=s+a[k][i];
printf("%d",s);
```

5.

Varianta C/C++

Pentru intensiv

```
#include<string.h>
#include<iostream.h>
void main()
{
    char cuv[20];
    unsigned int i,l,este;
    do
    { cout<<"cuvantul:"; cin>>cuv;
      este=1;
```

Varianta PASCAL

```
s:=0;
for j:=1 to n div 2 do
begin
    if a[k,2*j] mod 2=1
        s:=s+ a[k,2*j];
    if a[k,2*j+1] mod 2=1
        s:=s+a[k,2*j+1];
end;
writeln(s);
```

s:=0;

```
for j:=1 to n do
    if a[k,j] mod 2=1
        s:=s+ a[k,j];
writeln(s);
```

Varianta PASCAL

```
const lit:set of char=['a'..'z'];
      voc:set of
      'a'..'z'=['a','e','i','o','u'];
var cuv:string[20];
    i,j:byte;
    este:boolean;
begin
    repeat
        write('cuvantul:');readln(cuv);
```

```

l=strlen(cuv);
for(i=0;i<l;i++)
    if(!(cuv[i]>=97 &&
cuv[i]<=122))
        este=0;
}
while (este==0);
for(i=0;i<l;i++)
    switch(cuv[i])
{case`a`:cuv[i]=cuv[i]-32;break;
 case`e`:cuv[i]=cuv[i]-32;break;
 case`i`:cuv[i]=cuv[i]-32;break;
 case`o`:cuv[i]=cuv[i]-32;break;
 case`u`:cuv[i]=cuv[i]-32;break;
}
cout<<cuv;
}

```

Pentru neintensiv

```

#include <string.h>
#include <iostream.h>
#include <ctype>
void main()
{
char cuv[20];
unsigned int i,l,este;
do
{ cout<<"cuvantul:"; cin>>cuv;
este=1;
l=strlen(cuv);
for(i=0;i<l;i++)
    if(!(cuv[i]>=97 &&
cuv[i]<=122)|| !(cuv[i]>=65 &&
cuv[i]<=90))
        este=0;
}
while (este==0);
for(i=0;i<l;i++)
    if(cuv[i]>=97 && cuv[i]<=122)
        cuv[i]=toupper(cuv[i]);
cout<<cuv;
}

```

```

este:=true;
l:=length(cuv);
for i:=1 to l do
    if not (cuv[i] in lit) then
        este:=false;
until este;
for i:=1 to l do
    if cuv[i] in voc then
        cuv[i]:=chr(ord(cuv[i])-32);
writeln(cuv);
end.

```

```

const lit:set of
    char=['a'..'z','A'..'Z'];
    lit_mici:set of
    char=['a'..'z'];
var cuv:string[20];
    i,j:byte;
    este:boolean;
begin
    repeat
        write('cuvantul:');readln(cuv);
        este:=true;
        l:=length(cuv);
        for i:=1 to l do
            if not (cuv[i] in lit) then
                este:=false;
        until este;
        for i:=1 to l do
            if cuv[i] in lit_mici then
                cuv[i]:=chr(ord(cuv[i])-32);
        writeln(cuv);
    end.

```

SUBIECTUL III

1. d
2. 5310135 pentru **intensiv** și 55110 pentru **neintensiv**
- 3.

Varianta C/C++

Pentru intensiv

```

#include "stdio.h"
void main()
{ int n,a[100],x,i,aux,inv;
FILE *f;
f=fopen("NR.TXT","r");
n=0;

while (!feof(f))
{ fscanf(f,"%d",&x);
    if (x>0)
        { a[n]=x;

```

Varianta PASCAL

```

var f:text;
    a:array[1..100] of integer;
    n,i,x,aux:integer;
    inv:boolean;
begin
    assign(f,'NR.TXT');

    reset(f);
    n:=0;
    while not (eof(f)) do
        begin

```

```

        n++;
    }
}
fclose(f);
if (n==0)
    printf("NU EXISTA");
else
    { do
        { inv=0;
          for(i=0;i<n-1;i++)
            if (a[i])>a[i+1])
                { aux=a[i];
                  a[i]=a[i+1];
                  a[i+1]=aux;
                  inv=1;
                }
          }
        while (inv);

        for(i=0;i<n;i++)
            printf("%d",a[i]);
    }
}

```

4. a)

Varianta C/C++

```

unsigned f(unsigned a)
{ unsigned i,nr,s=0;
  i=2;
  while (i<=a && a!=0)
  { nr=0;
    while (a%i==0 && a!=0)
    { a=a/i;
      nr++;
    }
    if (nr!=0) s+=nr;
    else i++;
  }
  return s;
}

```

b)

Varianta C/C++

Pentru intensiv

```

#include<iostream.h>
unsigned
n,v[3],p[3],c,extraprim,aux;
unsigned f(unsigned a)
{ unsigned i,nr,s=0;

  i=2;
  while (i<=a && a!=0)
  { nr=0;
    while (a%i==0 && a!=0)

```

```

        read(f,x);
        if x>0 then
            begin
                n:=n+1;
                a[n]:=x;
            end;
        end;
        close(f);
        if n=0 then
            writeln('NU EXISTA')
        else
            begin
                repeat
                    inv:=false;
                    for i:=1 to n-1 do
                        if a[i]>a[i+1] then
                            begin
                                aux:=a[i];
                                a[i+1]:=a[i+1];
                                a[i]:=aux;
                                inv:=true;
                            end;
                until not inv;
                for i:=1 to n do
                    write(a[i],' ');
                end;
            end.

```

Varianta PASCAL

```

function f(a:word):word;
var i,nr,s:word;
begin
  s:=0;
  i:=2;
  while (i<=a) and (a<>0) do
  begin
    nr:=0;
    while (a mod i=0) and (a<>0) do
    begin
      a:=a div i;
      nr:=nr+1;
    end;
    if nr<>0 then s:=s+nr;
    else i:=i+1;
  end;
  f:=s;
end;

```

Varianta PASCAL

```

type vector=array[1..3] of word;
var v,p:vector;
    n,m,c,aux:word;
    extraprim:boolean;
function f(a:word):word;
var i,nr,s:word;
begin
  s:=0;
  i:=2;

```

```

    { a=a/i;
      nr++;
    }
    if (nr!=0) s+=nr;
    else i++;
  }
  return s;
}
void permut(unsigned i)
{ unsigned j,l,nr;
  if (i==c)
    {nr=0;
     for(l=0;l<c;l++)
nr=nr*10+v[p[l]];
    if
      (f(nr)>1)extraprim=0;
    }
  else
    { for(j=i;j<=i;j++)
      { aux=p[i];
        p[i]=p[j];
        p[j]=aux;
        permut(i+1);
        aux=p[i];
        p[i]=p[j];
        p[j]=aux;
      }
    }
}
void main()
{
  do
  {
    cout<<"n="; cin>>n;
  }
  while (n<2 || n>999);
  unsigned m=n;
  c=0;
  while (m!=0)
  { v[c]=m%10;
    m=m/10;
    c++;
  }
  if (c==3)
{aux=v[0];v[0]=v[2];v[2]=aux;}
  else if (c==2)
{aux=v[0];v[0]=v[1];v[1]=aux;}
  extraprim=1;
  permut(0);
  if (extraprim) cout<<"DA";
  else cout<<"NU";
}

```

```

while (i<=a) and (a<>0) do
begin
  nr:=0;
  while (a mod i=0) and (a<>0) do
  begin
    a:=a div i;
    nr:=nr+1;
  end;
  if nr<>0 then s:=s+nr
  else i:=i+1;
end;
f:=s;
end;
procedure permut(i:word);
var j,l,nr:word;
begin
  if i=c+1 then
  begin
    for j:=1 to c do
      nr:=nr*10+v[p[j]];
    if f(nr)>1 then
      extraprim:=false;
    end
  else
  begin
    p[i]:=i;
    for j:=1 to i do
    begin
      aux:=p[i];p[i]:=p[j];p[j]:=aux;
      permut(i+1);
      aux:=p[i];p[i]:=p[j];p[j]:=aux;
    end;
  end;
begin
  repeat
    write('n=');readln(n);
  until(n>=2) and (n<=999);
  m:=n;
  c:=0;
  while m<>0 do
  begin
    c:=c+1;
    v[c]:=m mod 10;
    m:=m div 10;
  end;
  if (c=3) then
  begin
    aux:=v[1];v[1]:=v[3];v[3]:=aux;
  end;
  else if (c=2) then
  begin
    aux:=v[1];v[1]:=v[2];v[2]:=aux;
  end;
  extraprim:=true;
  permut(1);
  if extraprim then write('DA')

```

Pentru neintensiv se modifică verificarea lui n , adică $n > 99$ în loc de $n > 999$ și dispăre `if(c==3)...` din funcția `main` rămânând doar `if(c==2)...`

```
else write('NU');
end.
Pentru neintensiv se modifică
verificarea lui n, adică  $n \leq 99$  în
loc de  $n \leq 999$ 
și dispăre if(c=3)... rămânând doar
if(c==2)...
```

Varianta <3>

SUBIECTUL I

1. b

2.

a) 599350

b) 197310

c) citește z, x (numere naturale nenule)

```
repetă
    citește y (număr natural)
    dacă  $z < y - x$  atunci
        scrie  $x \% 10$ 
    altfel
        scrie  $y \% 10$ 
    x ← y
până când  $x = 0$ 
```

d) Varianta C/C++

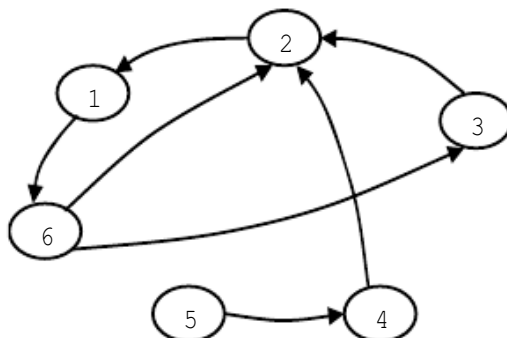
```
#include<iostream.h>
void main()
{
    unsigned int x,y,z;
    do
    {
        cout<<"z=";cin>>z;
        cout<<"x=";cin>>x;
    }
    while(z==0 || x==0);
    while(x>0)
    {
        cout<<"y=";cin>>y;
        if(z<y-x) cout<<x%10;
        else cout<<y%10;
        x=y;
    }
}
```

Varianta PASCAL

```
var x,y,z:word;
begin
    repeat
        write('\z=');readln(z);
        write('\x=');readln(x);
    until (x>0) and (z>0);
    while x>0 do
        begin
            write('y=');
            readln(y);
            if z<y-x then write(x mod 10)
                else write(y mod
10);
            x:=y;
        end;
    end.
```

SUBIECTUL II

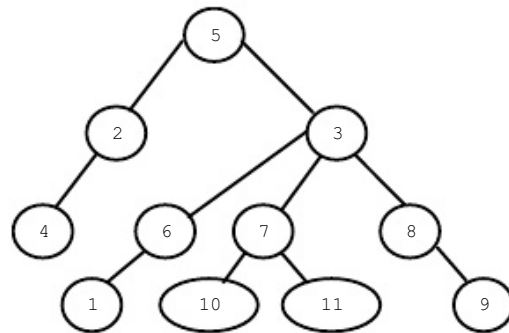
1. b



Pentru intensiv, drumul (5,4), (4,2), (2,1), (1,6), (6,3) are lungimea maximă5.
 Pentru neintensiv, nodurile 5 și 6 au gradul exterior strict mai mare decât cel interior.

2. c

1	2	3	4	5	6	7	8	9	10	11
6	5	5	2	0	3	3	3	8	7	7



Frunzele sunt: 4, 1, 10, 11, 9

3.

Varianta C/C++

```
f.x=f1.x*f2.y+f1.y*f2.x
f.y=f1.y*f2.y
```

Varianta PASCAL

```
f.x:=f1.x*f2.y+f1.y*f2.x
f.y:=f1.y*f2.y
```

4.

Șirul memorat în variabila s este **ideale** pentru intensiv

Șirul memorat în variabila s este **elen** pentru neintensiv

5.

Varianta C/C++

```
#include <iostream.h>
void main()
{
    unsigned n,A[24][24],i,j;
    do
    {
        cout<<"n="; cin>>n;
        while (n<2 || n>24);
        for(i=0;i<n;i++) A[i][i]=0;
        for(i=0;i<n;i++)
            for(j=0;j<n;j++)
                A[j][i]=n-i;
        for(i=0;i<n;i++)
            { for(j=0;j<n;j++)
                cout<<A[i][j]<<' ';
              cout<<'\n';
            }
    }
}
```

Varianta PASCAL

```
var A:array[1..24,1..24] of byte;
    n,i,j:byte;
begin
    repeat
        write('n=');readln(n);
    until (n>=2) and (n<=24);
    for i:=1 to n do A[i,i]:=0;
    for j:=1 to n do
        A[j,i]:=n-i;
    for i:=1 to n do
        begin
            for j:=1 to n do
                write(A[i,j],' ');
            writeln;
        end;
end.
```

SUBIECTUL III

1. a

2. xyyy pentru **intensiv** și xxxyyy pentru **neintensiv**

3.

Varianta C/C++

```
#include "stdio.h"
void main()
{
    int n,a[100],x,i,aux,inv;
    FILE *f;

    f=fopen("nr.txt","r");
    n=0;
    while (!feof(f))
    {
```

Varianta PASCAL

```
var f:text;
    a:array[1..100] of integer;
    n,i,x,aux:integer;
    inv:boolean;
begin
    assign(f,'NR.TXT');
    reset(f);
    n:=0;
    while not (eof(f)) do
```

```

    fscanf(f,"%d",&x);
    if (x>=100)
        {a[n]=x; n++;
        }
    }
fclose(f);
if (n==0)
    printf("NU EXISTA");
else
    { inv=0;
      for(i=0;i<n-1;i++)
          if (a[i]>a[i+1])
              { aux=a[i];
                a[i]=a[i+1];
                a[i+1]=aux;
                inv=1;
              }
    }

while (i<n; i++)
    printf("%d ",a[i]);
}

```

4. a)

Varianta C/C++

```

int cif(long a, int b)
{
    int nr=0;
    while (a!=0)
    {
        if (a%10==b) nr++;
        a=a/10;
    }
    return nr;
}

```

b)

Varianta C/C++

Pentru intensiv

```

#include <iostream.h>
long n,numar;
int cif(long a, int b)
{
    int nr=0;
    while (a!=0)
    {
        if (a%10==b) nr++;
        a=a/10;
    }
    return nr;
}
void main()
{

```

```

begin
    read(f,x);
    if x>=100 then
        begin
            n:=n+1;
            a[n]:=x;
        end;
    end;
close(f);
if n=0 then
    writeln('NU EXISTA')
else
    begin
        repeat
            inv:=false;
            for i:=1 to n-1 do
                if a[i]>a[i+1] then
                    begin
                        aux:=a[i];
                        a[i]:=a[i+1];
                        a[i+1]:=aux;
                        inv:=true;
                    end;
            until not inv;
            for i:=1 to n do
                write(a[i],' ');
            end;
        end.

```

Varianta PASCAL

```

function
cif(a:longint;b:integer):integer;
var nr:integer;
begin
    nr:=0;
    while (a<>0) do
        begin
            if (a mod 10=b) then nr:=nr+1;
            a:=a div 10;
        end;
    cif:=nr;
end;

```

Varianta PASCAL

```

Var x:array[1..8] of integer;
n,numar:longint;
i,j,p,t:integer;
ok:boolean;
function
cif(a:longint;b:integer):integer;
var nr:integer;
begin
    nr:=0;
    while (a<>0) do
        begin
            if (a mod 10=b) then nr:=nr+1;
            a:=a div 10;
        end;

```

```

do
{
  cout<<"n="; cin>>n;
} while (n<10000000 ||
n>99999999);
int i,ok=1,j,p=0,x[8],t;
for(i=9;i>=0;i--)
{t=cif(n,i);
  if (t!=0)
    if (t%2!=0)ok=0;
    else
      for(j=1;j<=t;j++)
        { x[p]=i;
          p++;
        }
}
if(ok)
{i=0;

  for(j=i+1;j<8-i;j++)
    x[j]=x[j+1];
  x[7-i]=x[i];
  i=i+1;
} while (i<=2);
numar=0;
for(i=0;i<8;i++)
  numar=numar*10+x[i];
cout<<numar;
}
else cout<<0;
}

```

Pentru neintensiv

```

#include <iostream.h>
long n;int i;
int cif(long a, int b)
{
  int nr=0;
  while (a!=0)
  {
    if (a%10==b) nr++;
    a=a/10;
  }
  return nr;
}
void main()
{
  do
  {
    cout<<"dati n(cu cel putin o
cifra impara)="; cin>>n;
  } while (n<0 || n>99999999);
  int j,t,numar=0;

```

```

  cif:=nr;
end;
begin
  repeat
    write('n=');readln(n);
  until
(n>=10000000)and(n<=99999999);
  ok:=true;
  p:=0;
  for i:=9 downto 0 do
  begin
    t:=cif(n,i);
    if t<>0 then
      if t mod 2<>0
        then ok:=false
      else
        for j:=1 to t do
          begin
            x[p]:=i;
            p:=p+1;
          end;
        end;
    if ok then
      begin
        i:=1;
        repeat
          for j:=i+1 to 9-i do
            x[j]:=x[j+1];
          x[9-i]:=x[i];
          i:=i+1;
        until i>3;
        numar:=0;
        for i:=1 to 8 do
          numar:=numar*10+x[i];
        writeln(numar);
      end
    else writeln(0);
  end.

```

```

var x:array[1..8] of integer;
n:longint;
i,j,t,numar:integer;
function
cif(a:longint;b:integer):integer;
var nr:integer;
begin
  nr:=0;
  while (a<>0) do
  begin
    if (a mod 10=b) then nr:=nr+1;
    a:=a div 10;
  end;
  cif:=nr;
end;
begin
  repeat
    write('dati n(cu cel putin o
cifra impara)=');readln(n);
  until (n>=0) and (n<=99999999);
  numar:=0;

```



```

for(i=9;i>=1;i=i-2)
{t= cif(n,i);
 if (t!=0)
  for(j=1;j<=t;j++)
numar=numar*10+i;
}
if(numar==0) cout<<"numarul nu are cifre impare";
else cout<<numar;
}

```

```

i:=9;
while i>=1 do
begin
  t:=cif(n,i);
  if (t<>0) then
    for j:=1 to t do
      numar:=numar*10+i;
  i:=i-2;
end;
if numar=0 then write('numarul nu are cifre impare')
else writeln(numar);
end.

```

Varianta <4>

SUBIECTUL I

1. d

2.

a) 16 14 12 10 8 6

b) perechile de valori care pot fi citite pentru a și b sunt:

-10, 0

0, -10

-11, 0

0, -11

-10, 1

1, -10

-11, 1

1, -11

c) citește a,b (numere întregi)

```

dacă a<b atunci
  s←a; a←b; b←s;

```

```

x←a
cât timp x>=b execută
  dacă x%2=0 atunci
    scrie x, ' '
  x←x-1

```

d) Varianta C/C++

```

#include<iostream.h>
void main()
{
  int a,b,x,s;
  cout<<"a=";cin>>a;
  cout<<"b=";cin>>b;
  if (a<b)
  {
    s=a; a=b; b=s;
  }
  for(x=a;x=b;x--)
  if(x%2==0)
    cout<<x<<' ';
}

```

Varianta PASCAL

```

var a,b,x,s:integer;
begin
  write('a=');readln(a);
  write('b=');readln(b);
  if a<b then
  begin
    s:=a;a:=b;b:=s;
  end;
  for x=a downto b do
    if x%2=0 then
      write(x,' ');
end.

```

SUBIECTUL II

1. b

2. a

3.

2	vârf
1	bază

4.

Varianta C/C++

Pentru intensiv

```
if (s[i]>=97 && s[i]<=122)
{ for(j=i;j<strlen(s)-1;j++)
    s[j]=s[j+1];
  s[strlen(s)-1]='\0';
}
else i++;
```

Pentru neintensiv

```
if (s[i]>=97 && s[i]<=122)
  s[i]='W';
i++;
```

5.

Varianta C/C++

```
#include<iostream.h>
void main()
{ unsigned a[24],n,i,j;
  do
  {
    cout<<"n="; cin>>n;
    while (n<2||n>24);
    for(i=0;i<n;i++) a[i][n-1-i]=0;
    for(i=0;i<n;i++)
      for(j=0;j<n;j++)
        if (j!=n-1-i)
          a[i][j]=n-i;
    for(i=0;i<n;i++)
      { for(j=0;j<n;j++)
          cout<<a[i][j]<<' ';
        cout<<'\\n';
      }
  }
}
```

Varianta PASCAL

```
if (s[i]>='a') and (s[i]<='z') then
  delete(s,i,1)
else
  i:=i+1;
```

```
if (s[i]>='a' and s[i]<='z') then
  s[i]:='W';
i:=i+1;
```

Varianta PASCAL

```
var a:array[1..24,1..24] of byte;
    n,i,j:byte;
begin
  repeat
    write('n=');readln(n);
  until (n>=2) and (n<=24);
  for i:=1 to n do
    for j:=1 to n do
      if j<>n+1-i then
        a[i,j]:=n-i;
  for i:=1 to n do
    begin
      for j:=1 to n do
        write(a[i,j],' ');
      writeln;
    end;
end.
```

SUBIECTUL III

1. c

2. 01111 pentru intensiv și *4062 pentru neintensiv

3.

Varianta C/C++

pentru intensiv

```
#include "stdio.h"
void main()
{ int n,a[100],x,i,aux,inv;
  FILE *f;
  f=fopen("NR.TXT","r");
  n=0;
  while (!feof(f))
```

Varianta PASCAL

```
var f:text;
    a:array[1..100] of integer;
    n,i,x,aux:integer;
    inv:boolean;
begin
  assign(f,'NR.TXT');
  reset(f);
```

```

{
  fscanf(f, "%d", &x);
  if (x<100) a[n++]=x;
}
fclose(f);
if (n==0)
  printf("NU EXISTA");
else
  { //ordonam vectorul descrescator
  do
  { inv=0;
    for(i=0;i<n-1;i++)
      if (a[i]<a[i+1])
        { aux=a[i];
          a[i]=a[i+1];
          a[i+1]=aux;
          inv=1;
        }
    } while (!inv);
  }
  for(i=0;i<n;i++)
    printf("%d ",a[i]);
}

```

4. a)

Varianta C/C++

```

int cif(long a, int b)
{
  int nr=0;
  while (a!=0)
  {
    if (a%10==b) nr++;
    a=a/10;
  }
  return nr;
}

```

b)

Varianta C/C++

Pentru intensiv

```

#include <iostream.h>
long n,numar;
int cif(long a, int b)
{
  int nr=0;
  while (a!=0)
  {
    if (a%10==b) nr++;
    a=a/10;
  }
  return nr;
}

```

```

n:=0;
while not (eof(f)) do
begin
  read(f,x);
  if x<100 then
  begin
    n:=n+1;
    a[n]:=x;
  end;
end;
close(f);
if n=0 then
  writeln('NU EXISTA')
else
begin
  repeat
    inv:=false;
    for i:=1 to n-1 do
      if a[i]<a[i+1] then
        begin
          aux:=a[i];
          a[i]:=a[i+1];
          a[i+1]:=aux;
          inv:=true;
        end;
    until not inv;
    for i:=1 to n do
      write(a[i], ' ');
    end;
end.

```

Varianta PASCAL

```

function
cif(a:longint;b:integer):integer;
var nr:integer;
begin
  nr:=0;
  while (a<>0) do
  begin
    if (a mod 10=b) then nr:=nr+1;
    a:=a div 10;
  end;
  cif:=nr;
end;

```

Varianta PASCAL

```

Var x:array[1..8] of integer;
    n,m,numar:longint;
    i,j,p,t:integer;
    ok:boolean;
function
cif(a:longint;b:integer):integer;
var nr:integer;
begin
  nr:=0;
  while (a<>0) do
  begin
    if (a mod 10=b) then nr:=nr+1;

```

```

void main()
{
  int i,ok=1,j,p=0,x[8],t;
  do
  {
    cout<<"n="; cin>>n;
    long m=n;
    while (m!=0 && ok)
    {if(m%10==0) ok=0;
     m=m/10;}
  }
  while ((n<10000000 ||
n>99999999)||
         ok==0);
  for(i=1;i<=9;i++)
  {t=cif(n,i);
  if (t!=0)
    if (t%2!=0)ok=0;

    else
      for(j=1;j<=t;j++)
        { x[p]=i;
          p++;
        }
  }
  if(ok)
  {i=0;
  do
  { for(j=i+1;j<8-i;j++)
    x[j]=x[j+1];
    x[7-i]=x[i];
    i=i+1;
  } while (i<=2);
  numar=0;
  for(i=0;i<8;i++)
    numar=numar*10+x[i];
  cout<<numar;
  }
  else cout<<0;
}

```

Pentru neintensiv

```

#include <iostream.h>
long n;int i;
int cif(long a, int b)
{
  int nr=0;
  while (a!=0)
  {
    if (a%10==b) nr++;
    a=a/10;
  }
  return nr;
}

```

```

    a:=a div 10;
  end;
  cif:=nr;
end;
begin
  repeat
    write('n=');readln(n);
    m:=n;ok:=true;
    while (m<>0) and ok do
    begin
      if m%10=0 then ok:=false;
      m:=m/10;
    end;
  until
  ((n>=10000000)and(n<=99999999))and
ok;
  ok:=true;
  p:=0;

  for i:=1 to 9 do
  begin
    t:=cif(n,i);
    if t<>0 then
      if t mod 2<>0
        then ok:=false
      else
        for j:=1 to t do
        begin
          p:=p+1;
          x[p]:=i;
        end;
      end;
  end;
  if ok then
  begin
    i:=1;
    repeat
      for j:=i+1 to 9-i do
        x[j]:=x[j+1];
        x[9-i]:=x[i];
      i:=i+1;
    until i>3;
    numar:=0;
    for i:=1 to 8 do
      numar:=numar*10+x[i];
    writeln(numar);
  end
  else writeln(0);
end.

```

```

var x:array[1..8] of integer;
    n:longint;
    i,j,t,numar:integer;
function
cif(a:longint;b:integer):integer;
var nr:integer;
begin
  nr:=0;
  while (a<>0) do
  begin
    if (a mod 10=b) then nr:=nr+1;
    a:=a div 10;
  end;

```

```
void main()
{
    do
    {
        cout<<"dati n(cu cel putin o cifra
            impara)="; cin>>n;
    } while (n<0 || n>999999999);
    int j,t,numar=0;
    for(i=1;i<=9;i=i+2)
    {t= cif(n,i);
    if (t!=0)
        for(j=1;j<=t;j++)
            numar=numar*10+i;
    }
    if(numar==0) cout<<"numarul nu are cifre impare";
    else cout<<numar;
}

```

```
end;
cif:=nr;
end;
begin
    repeat
        write('dati n(cu cel putin o cifra impara)=');readln(n);
    until (n>=0) and (n<=999999999);
    numar:=0;
    i:=1;
    while i<=9 do
    begin
        t:=cif(n,i);
        if (t<>0) then
            for j:=1 to t do
                numar:=numar*10+i;
            i:=i+2;
        end;
    end;
    if numar=0 then write('numarul nu are cifre impare');
    else writeln(numar);
end.

```

Varianta <5>

SUBIECTUL I

1. c

2.

a) 1

b) 890 numere

c) citește x, z (numere naturale)

```
y ← 0
cât timp x > 0 execută
    y ← y * 10 + x % 10
    x ← [x / 100]
dacă y = z atunci
    scrie 1
altfel
    scrie 0

```

d) Varianta C/C++

```
#include<iostream.h>
void main()
{
    long int x,y,z;
    cout<<"x=";cin>>x;
    cout<<"z=";cin>>z;
    y=0;
    do
    {
        y=y*10+x%10;
        x=x/100;
    }
    while (x>0);
    while (y*z>0 && y%10==z%10)
    {

```

Varianta PASCAL

```
var x,y,z:longint;
begin
    write('x=');readln(x);
    write('z=');readln(z);
    y:=0;
    repeat
        y:=y*10+x mod 10;
        x:=x div 100
    until x=0;
    while (y*z>0) and (y mod 10= z mod 10)
    do
    begin
        y:=y div 10;
        z:=z div 10;
    end;

```

```

    y=y/10;
    z=z/10;
}
if (y+z==0)
    cout<<1;
else
    cout<<0;
}

```

```

if y+z=0 then
    write(1)
else
    write(0)
end.

```

SUBIECTUL II

1. b

2. d

3. 14

4. Varianta C/C++

Pentru intensiv

```

c=s[i];
s[i]=s[j];
s[j]=c;
i++;
j--;

```

Pentru neintensiv

```

if (s[i]= =c) s[i]=d;

```

5.

Varianta C/C++

```

#include <iostream.h>
void main()
{ unsigned A[11][11],n,m,i,j;
  do
  { cout<<"n="; cin>>n;
  }
  while (n<2 || n>10);
  do
  {
    cout<<"m="; cin>>m;
  }
  while (m<2 || m>10);
  for(i=1;i<=n;i++)
    for(j=1;j<=m;j++)
      if (i>j) A[i][j]=i;
      else A[i][j]=j;
  for(i=1;i<=n;i++)
  {
    for(j=1;j<=m;j++)
      cout<<A[i][j]<<' ';
    cout<<' \n';
  }
}

```

Varianta PASCAL

```

c:=s[i];
s[i]:=s[j];
s[j]:=c;
i:=i+1;
j:=j-1;

```

```

if s[i]=c then s[i]:=d;

```

Varianta PASCAL

```

var n,m,,i,j:byte;
    A:array[1..10,1..10] of byte;
begin
  repeat
  until (n>=2) and (n<=10);
  repeat
    write('m=');readln(m);
  until (m>=2) and (m<=10);
  for i:=1 to n do
    for j:=1 to m do
      if i>j then A[i,j]:=i
      else A[i,j]:=j;
  for i:=1 to n do
  begin
    for j:=1 to m do
      write(A[i,j], ' ');
    writeln;
  end
end.

```

SUBIECTUL III

1. d

2. 7***** pentru intensiv și 2***** pentru neintensiv

3. **Varianta C/C++**

pentru intensiv
#include <stdio.h>

Varianta PASCAL

```

var f:text;

```

```
void main()
{ long n;
  FILE *f;
  f=fopen("NR.TXT","w");
  do
  { printf("n=";
    scanf("%d",&n);
  } while (n<10 || n>99999999);
  while (n!=0)
  { fprintf(f,"%d ",n);
    n=n/10;
  }
  fclose(f);
}
```

4. a)

Varianta C/C++

```
unsigned long f(unsigned long a)
{ unsigned long d=2;
  while (a%d) d++;
  return d;
}
```

b)

Varianta C/C++

Pentru intensiv

```
#include <iostream.h>
unsigned long x[100],y[100];
unsigned n;
unsigned long f(unsigned long a)
{ unsigned long d=2;
  while (a%d) d++;
  return d;
}
void main()
{ unsigned i,m,inv;
  unsigned long aux;
  do
  { cout<<"n=";
    cin>>n;
  }
  while (n<1 || n>100);

  for(i=0;i<n;i++)
  { cout<<"x["<<i<<"]=";
    cin>>x[i];
  }
  while (x[i]<2 || x[i]>99999999);
  m=0;
  for(i=0;i<n;i++)
  if (f(x[i])==x[i])
    y[m++]=x[i];
  if (m==0)
    cout<<"NU EXISTA";
  else
  { //ordonarea vectorului
    do
    { inv=0;
      for(i=0;i<m-1;i++)
```

```
n:longint;
begin
  assign(f,'NR.TXT');
  rewrite(f);
  repeat
    write('n=');readln(n);
  until (n>=10) and (n<=99999999);
  while n<>0 do
  begin
    write(f,n,' ');
    n:=n div 10;
  end;
  close(f);
end.
```

Varianta PASCAL

```
function f(a:longint):longint;
var d:longint;
begin
  d:=2;
  while(a mod d<>0) d:=d+1;
  f:=d;
end;
```

Varianta PASCAL

```
type vector=array[1..100] of
longint;
var y:vector;
    n,m,i:integer;
    x,aux:longint;
    inv:boolean;

function f(a:longint):longint;
var d:longint;
begin
  d:=2;
  while(a mod d<>0) do d:=d+1;
  f:=d;
end;
begin
  repeat
    write('n=');readln(n);
  until (n>=10) and (n<=99999999);
  m:=0;
  for i:=1 to n do
  begin
    repeat
      write('x=');
      readln(x);
    until (x>1) and (x<=99999999);
    if f(x)=x then
    begin
      m:=m+1;
      y[m]:=x;
    end;
  end;
  if m=0 then write('NU EXISTA')
else
```

```

        if (y[i]>y[i+1])
        { aux=y[i];
          y[i]=y[i+1];
          y[i+1]=aux;
          inv=1;
        }
    } while (inv);
    //afisarea numerelor obtinute
    for (i=0;i<m;i++)
        cout<<y[i]<<' ';
    }
}

```

```

begin
{ordonarea vectorului}
repeat
    inv:=false;
    for i:=1 to m-1 do
        if y[i]>y[i+1] then
            begin
                aux:=y[i];
                y[i]:=y[i+1];
                y[i+1]:=aux;
                inv:=true;
            end;
    until not inv;
    {afisarea numerelor obtinute}
    for i:=1 to m do
        write(y[i],' ');
    end;
end.

```

Pentru neintensiv

Program asemănător cu cel de la varianta intensivă dar fără secvența de ordonare.

Varianta <6>

SUBIECTUL I

1. a

2.

a) 9

b) 13

c)

```

citește n (număr natural)
s ← -1
dacă n > 0 atunci
    repetă
        dacă n % 10 > s atunci
            s ← n % 10
        altfel
            s ← 11
        n ← [n / 10]
    până când n = 0

```

scrie s

d) Varianta C/C++

```

#include<iostream.h>
void main()
{
    unsigned int n,s;
    cout<<"n=";cin>>n;
    s=-1;
    while (n>0)
    {
        if (n%10>s)
            s=n%10;
        else
            s=11;
        n=n/10;
    }
}

```

Varianta PASCAL

```

var n,s:word;
begin
    write('n=');readln(n);
    s:=-1;
    while n>0 do
        begin
            if n mod 10>s then
                s:=n mod 10
            else
                s:=11;
            n:=n div 10;
        end;
    write(s)
end.

```



```
    cout<<s;
}
```

SUBIECTUL II

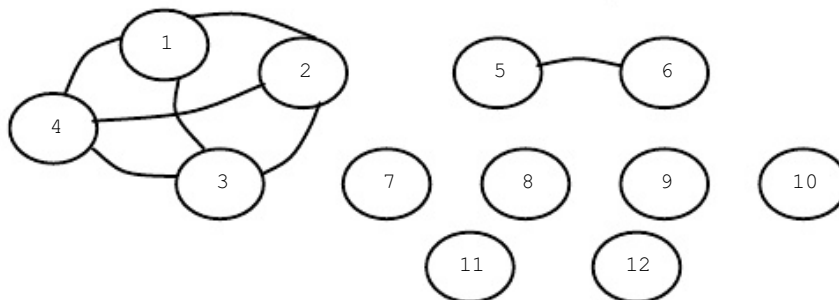
1. c pentru **intensiv** și a pentru **neintensiv**

2. b pentru **intensiv** și c pentru **neintensiv**

3. 12 noduri pentru **intensiv** și 14 muchii pentru **neintensiv**

4. Pentru **intensiv**

8 componente conexe



Pentru **neintensiv**

16

5.

Varianta C/C++

```
#include <iostream.h>
#include <string.h>
#include <ctype.h>
void main()
{ char t[255];
  unsigned i,l,p1=1;
  cout<<"text:";
  cin.get(t,256);
  l=strlen(t);
  for(i=0;i<l;i++)
    if (p1 && t[i]!=' ')
      { t[i]=toupper(t[i]);
        p1=0;
      }
    else
      if (t[i]==' ')
        p1=1;
  cout<<t;
}
```

Varianta PASCAL

```
Var t:string;
    i,l:word;
    pl:boolean;
begin
  write('text:');readln(t);
  l:=length(t);
  pl:=true;
  for i:=1 to l do
    if pl and (t[i]<>' ') then
      begin
        t[i]:=chr(ord(t[i])-32);
        pl:=false;
      end
    else
      if t[i]=' ' then pl:=true;
  write(t)
end.
```

SUBIECTUL III

1. a

2. 7 pentru **intensiv** și 143 pentru **neintensiv**

3.

Varianta C/C++

Pentru **intensiv**

```
#include <iostream.h>
void main()
{ unsigned a[100],n,i,s=0;
  do
    { cout<<"n=";
      cin>>n;
```

Varianta PASCAL

```
type vector=array[1..100] of word;
var a:vector;
    n,i,s:word;
begin
  repeat
    write('n=');
```

```

}
while (n<2 || n>100);
for(i=0;i<n;i++)
do
{ cout<<"a["<<i<<"]=";
cin>>a[i];
}
while (a[i]>9999);
for(i=0;i<n;i++)
s=s+a[i];
cout<<s<<endl;
for(i=1;i<n;i++)
{
s=s-a[n-i];
cout<<s<<endl;
}
}

```

Pentru neintensiv

a) void s1(int &a,int &b)

b)

```

int s2(int a[100],int p,int q)
{ int i=p;
if(p<q)
{while (i<=q && a[i]%5!=0) i++;
if (i<=q) return i;
else return -1;
}
else
{while(i>=q&& a[i]%5!=0) i--;
if(i>=q) return i;
else return -1;
}
}

```

c)

```

#include <fstream.h>
ofstream f("bac.txt");

```

```

int a[100],n,p1,p2,i,b)
void s1(int &a,int &b)
{int c;
c=a;a=b;b=c;
}
int s2(int a[100],int p,int q)
{ int i=p ;
if(p<q)
{while (i<=q && a[i]%5!=0) i++;
if (i<=q) return i;
else return -1;
}
else
{ while (i>=q && a[i]%5!=0) i--;
if (i>=q) return i;
else return -1;
}
}

```

```

readln(n);
until (n>=1)and(n<=100);
for i:=1 to n do
repeat
write('a[' ,i,']=');
readln(a[i]);
until (a[i]<=9999);
for i:=1 to n do s:=s+a[i];
writeln(s);
for i:=2 to n do
begin
s:=s-a[n+2-i];
writeln(s);
end;
end.

```

a) procedure s1(var a:integer;var b:integer);

b)

```

function
s2(a:vector;p,q:integer):integer;
var i:integer;
begin
i:=p;
if(p<q)
begin
while (i<=q) and (a[i] mod 5<>0)
do
i:=i+1;
if i<=q then s2:=i
else s2:=-1;
end
else
begin
while (i>=q) and (a[i] mod 5<>0)
do
i:=i-1;
if i>=q then s2:=i
else s2:=-1;
end;
end;

```

c)

```

type vector=array[1..100] of integer;
var a:vector;
n,p1,p2,i:integer;
procedure s1(var a:integer;var b:integer);
var c:integer;
begin
c:=a;a=b;b=c;
end;
function
s2(a:vector;p,q:integer):integer;
var i:integer;
begin
i:=p;
if(p<q) then

```

```

}
void main( )
{ do
  { cout<<'n=';cin>>n;
  }
  while (n<1 || n>=100);
  for(i=0;i<n;i++)
  do
  { cout<<'a['<<i<<']=';
  cin>>a[i];
  }
  while (a[i]<-9999||a[i]>9999);
  p1=s2(a,0,n-1);p2=s2(a,n-1,0);
  if(p1==-1||p2==-1) f<<0;
  else
  {s1(a[p1],a[p2]);
  for(i=0;i<n;i++) f<<a[i]<<" ";
  }
f.close();

```

4. a)

Varianta C/C++

```

#include "stdio.h"

void main()
{ long n1,n2,nr_ap;
  FILE *f;
  f=fopen("BAC.TXT","r");
  fscanf(f,"%ld",&n1);
  nr_ap=1;
  while(!feof(f))
  { fscanf(f,"%ld",&n2);
    if (n1==n2) nr_ap++;
    else
    { printf("%ld %ld ",n1,nr_ap);
      nr_ap=1;
      n1=n2;
    }
  }
  printf("%ld %ld ",n1,nr_ap);

```

```

begin
  while (i<=q) and (a[i] mod 5<>0)
do
  i:=i+1;
  if i<=q then s2:=i
  else s2:=-1;
end
else
begin
  while (i>=q) and (a[i] mod 5<>0)
do
  i:=i-1;
  if i>=q then s2:=i
  else s2:=-1;
end;
end;
begin
repeat
  write('n=');
  readln(n);
until (n>=1)and(n<=100);
for i:=1 to n do
  repeat
  write('a[' ,i,']=');
  readln(a[i]);
  until (a[i]>=-9999)and(a[i]<=9999);
  p1:=s2(a,1,n);p2:=s2(a,n,1);
  assign(f,'Bac.txt');
  rewrite(f);
  if (p1=-1) or (p2=-1) then
  write(f,0)
  else
  begin
  s1(a[p1],a[p2]);
  for i:=1 to n do write(f,a[i],
  ');
  end;
close(f);
end.

```

Varianta PASCAL

```

var f:text;
n1,n2,nr_ap:longint;
begin
  assign(f,'BAC.TXT');
  reset(f);
  read(f,n1);
  nr_ap:=1;
  while not(eof(f)) do
  begin
  read(f,n2);
  if n1=n2 then nr_ap:=nr_ap+1
  else
  begin
  write(n1,' ',nr_ap,' ');
  nr_ap:=1;
  n1:=n2;
  end;
end;

```

```

}
end;
write(n1, ' ', nr_ap, ' ');
end.

```

b) Se citește primul număr din fișier și se inițializează variabila **nr_ap** cu valoarea 1. Cât timp nu s-a întâlnit sfârșitul de fișier se mai citește un nr și se compară cu cel anterior citit. Dacă sunt egale crește cu 1 numărul de apariții; dacă nu sunt egale se tipărește primul dintre ele cu numărul de apariții corespunzător, se reinițializează variabila care numără aparițiile cu valoarea 1, după care se reconsideră primul număr din secvența de numere egale ca fiind cel de-al 2-lea citit.

Varianta <7>

SUBIECTUL I

1. d.

2.

a) 9432

b) 4569, 45690

c) citește n (număr natural nenul)

```

nr←0; a←9
repetă
┌   m←n
├   ┌cât timp m≠0 și m%10≠a execută
├   │   m←[m/10]
├   └─┘
├   ┌dacă m≠0 atunci
├   │   nr←nr*10+m%10
├   └─┘
└   a←a-1
până când a<0
scrie nr

```

d) Varianta C/C++

```

#include<iostream.h>
void main()
{
    long int n,nr,m,a;
    do
    {
        cout<<"n=";<<cin>>n;
    }

    while (n<1);
    for (a=9;a>=0;a--)
    {
        m=n;
        while (m!=0 && m%10!=a)
        {
            m=m/10;
        }
        if (m!=0)
            nr=nr*10+m%10;
    }
    cout<<nr;
}

```

Varianta PASCAL

```

var n,nr,m,a:longint;
begin
    repeat
        write('n=');readln(n);
    until n>0;
    nr:=0;
    for a=9 downto 0 do
        begin
            while (m<>0) and (m mod 10<>a)
            do
                m:=m div 10;
            if m<>0 then
                nr:=nr*10+m mod 10;
            end;
        write(nr)
    end.

```

SUBIECTUL II

1. a

2. b pentru intensiv și d pentru neintensiv

3. 2 pentru intensiv

Varianta C/C++

pentru neintensiv

```
cout<<x.nume<<x.clasa<<x.medie;
```

4. 128 pentru intensiv și 2 pentru neintensiv

5.

Varianta C/C++

pentru intensiv

```
#include <iostream.h>
#include <string.h>
void main()
{
    char t[255],v[]="aeiou ";
    int i,n,p=-1,ok;
    do
    {
        cin.get(t,256);
        ok=1;
        n=strlen(t);
        for(i=0;i<n;i++) //verificam daca e
//litera mica sau spatiu
            if ((t[i]<97 || t[i]>122) &&
t[i]!=32)
                ok=0;
            else
                if (strchr(v,t[i])==NULL)
//verificam daca e consoana
                    p=i;
    }
    while (!ok);
    for(i=p;i<n-1;i++)
        t[i]=t[i+1];
    t[n-1]=NULL;
    cout<<t ;
}
```

pentru neintensiv

```
#include <iostream.h>
#include <string.h>
void main()
{
    char t[255],v []="aeiou ";
    int i,n,p=-1,ok;
    do
    {
        cin.get(t,256);
        ok=1;
        n=strlen(t);
        for(i=0;i<n;i++) //verificam daca e
litera mica sau spatiu
            if ((t[i]<97 || t[i]>122) &&
t[i]!=32)
                ok=0;
            else
                if
(strchr(v,t[i])==NULL)//verificam
```

Varianta PASCAL

```
write(x.nume,x.clasa,x.medie);
```

Varianta PASCAL

```
const v: set of
'a'..'z'=['a','e','i','o','u'];
var t:string;
    n,p,i:integer;
    ok:boolean;

begin
repeat
    write('text:');
    readln(t);
    n:=length(t);
    ok:=true;
    for i:=1 to n do
        if((t[i]<'a') or (t[i]>'z')) and
            (t[i]<>' ') then ok:=false
        else
            if not(t[i] in v) then p:=i;
    until ok;
    delete(t,p,1);
    write(t);
end.
```

```
const v: set of
'a'..'z'=['a','e','i','o','u'];
var t:string;
    n,p,i:integer;
    ok:Boolean;

begin
repeat
    write('text:');
    readln(t);
    n:=length(t);
    ok:=true;
    for i:=1 to n do
        if((t[i]<'a') or (t[i]>'z')) and
            (t[i]<>' ') then ok:=false
        else
            if not(t[i] in v) then p:=i;
    until ok;
    write(t[p]);
end.
```

```
daca e consoana
    p=i;
}
while (!ok);
cout<<t[p];
}
```

SUBIECTUL III

1. c pentru intensiv și d pentru neintensiv

2. $C_9^5 = 126$ numere

3.

Varianta C/C++

pentru intensiv

```
#include "stdio.h"
void main()
{ unsigned v[300],n,i,exista,aux,p;
  FILE *f;
  f=fopen("BAC.TXT","w");
  do
  {
    printf("n=");
    scanf("%d",&n);
  }
  while (n<1 || n>100);
  for(i=0;i<3*n;i++)
  do
  {
    printf("v[%d]=",i);
    scanf("%d",&v[i]);
  }
  while (v[i]>9999);
  i=0; exista=0;
  while (i<n && v[i]%2!=0)
    i++;
  if (i<n)
    { exista=1;
      p=i;
    }
  if (exista)
  {
    i=3*n-1;

    exista=0;
    while (i>=2*n && v[i]%2==0)
      i--;
    if (i>=2*n)
    {exista=1;
      aux=v[p];
      v[p]=v[i];
      v[i]=aux;
    }
  }
  for(i=0;i<3*n;i++)
    fprintf(f,"%d ",v[i]);
  fclose(f);
}
```

Varianta PASCAL

```
type vector=array[1..30] of word;
var v:vector;
    n,i,aux,p:word;
    exista:boolean;
    f:text;
begin
  assign(f,'BAC.TXT');
  rewrite(f);
  repeat
    write('n=');readln(n);
  until (n>0)and(n<=100);
  for i:=1 to 3*n do
    repeat
      write('v[' ,i,']=');
      readln(v[i]);
    until v[i]<10000;
  i:=1;
  exista:=false;
  while (i<=n)and(v[i] mod 2<>0)
    do i:=i+1;
  if i<=n then
  begin
    exista:=true;p:=i;
  end;
  if exista then
  begin
    i:=3*n; exista:=false;
    while(i>2*n)and(v[i] mod 2=0)
      do (i:=i-1); then
    begin
      exista:=true;
      aux:=v[p];
      v[p]:=v[i];
      v[i]:=aux;
    end;
  end;
```

pentru intensiv

```
#include "stdio.h"
void main()
{ unsigned v[300],n,i,exista,aux,p;
  FILE *f ;
  f=fopen("BAC.TXT","w");
  do
  {
    printf("n=");
    scanf("%d",&n);
  }
  while (n<1 || n>100);
  for(i=0;i<3*n;i++)
  do
  {
    printf("v[%d]=",i);
    scanf("%d",&v[i]);
  }

  while (v[i]>9999);
  for(i=0;i<n;i++)
  {
    aux=v[i];
    v[i]=v[2*n+i];
    v[2*n+i]=aux;
  }
  for(i=0;i<3*n;i++)
  fprintf(f,"%d ",v[i]);
  fclose(f);
}
```

4.

**Varianta C/C++
pentru intensiv**

a)

```
unsigned long sub (unsigned long n)
{
  unsigned long t=5;
  if (n>5)
  { while (n>t) t=2*t;
    t=t/2;
  }
  else t=n;
  return t;
}
```

b)

```
#include <iostream.h>
unsigned long s,n;
unsigned long sub (unsigned long n)
{
  unsigned long t=5;
  if (n>5)
  { while (n>t) t=2*t;
    t=t/2;
  }
  else t=n;
  return t;
}
void main()
```

```
type vector=array[1..30] of word;
var v:vector;
    n,i,aux,p:word;
    exista:boolean;
    f:text;
begin
  assign(f,'BAC.TXT');
  rewrite(f);
  repeat
    write('n=');readln(n);
  until (n>0)and(n<=100);
  for i:=1 to 3*n do
    repeat
      write('v[' ,i,']=');
      readln(v[i]);
    until v[i]<10000;
  for i:=1 to n do
    begin
      aux:=v[i];
      v[i]:=v[2*n+i];
      v[2*n+i]:=aux;
    end;
  for i:=1 to 3*n do
    write(f,v[i],' ');
  close(f);
end.
```

Varianta PASCAL

```
function sub(n:longint):longint;
var t:longint;
begin
  if n>5 then
  begin
    while n>t do t:=2*t;
    t:=t div 2;
  end
  else t:=n;
end;
sub:=t;
```

```
var s,n:longint;
function sub(n:longint):longint;
var t:longint;
begin
  t:=5;
  if n>5 then
  begin
    while n>t do t:=2*t;
    t:=t div 2;
  end
  else t:=n;
  sub:=t;
end;
```

```

{
do
{ cout<<"s=";
cin>>s;
}
while (s>10000000);
while (s>0)
{
n=sub(s);
cout<<n<<' \';
s=s-n;
}
}
pentru neintensiv
a) unsigned long p(unsigned long n)
#include <iostream.h>
#include <fstream.h>
ofstream g("Numere.txt");

unsigned long sub(unsigned long n)
{
unsigned long t=5;
if (n>5)
{ while (n>t) t=2*t;
t=t/2;
}
else t=n;
return t;
}
void main()
{
do
{ cout<<"s=";
cin>>s;
}
while (s>10000000);
while (s>0)
{
n=sub(s);
g<<n<<' \';
s=s-n;
}
g.close();
}

```

```

begin
repeat
write('s=');readln(s);
until (s>=0) and (s<=10000000);
while (s>0) do
begin
n:=sub(s);
write(n,' ');
s:=s-n
end
end.

```

```

function p(n:longint):longint;
var s,n:longint;
f:text;
function sub(n:longint):longint;
var t:longint;
begin
t:=5;
if n>5 then
begin
while n>t do t:=2*t;
t:=t div 2;
end
else t:=n;
sub:=t;
end;
begin
assign(f,'Numere.txt');
rewrite(f);
repeat
write('s=');readln(s);
until (s>=0) and (s<=10000000);
while (s>0) do
begin
n:=sub(s);
write(f,n,' ');
s:=s-n;
end;
close(f);
end.

```

Varianta <8>

SUBIECTUL I

1. a

2. a) 22

b) Dacă șirul are ultimele k cifre impare atunci valoarea returnata va fi 0.

$n=571355$ și $k=4$

$n=572355$ și $k=3$

$n=1357$ și $k=78$

$n=ab^1b^2\dots b^k$ unde $b^1b^2\dots b^k$ sunt k cifre impare

c) citește n, k (numere naturale nenule)

$nr \leftarrow 0$


```

p ← 1
┌-execută
|   ┌- dacă n%2=0 atunci
|   |       nr ← nr+n%10*p
|   |       p ← p*10
|   |   altfel
|   |       k ← k-1
|   └-
|   ┌■
|       n ← [n/10]
|   cat timp
└■ cat timp n ≠ 0 și k ≠ 0
    
```

scrie nr

d) Varianta C/C++

```

#include<iostream.h>
#include<conio.h>

void main( )
{long int n,k,nr=0,p=1;
clrscr();
cin>>n;
cin>>k;
while (n!=0 && k!=0)

if (n%2==0)
    {nr=nr+n%10*p;
    p=p*10;
    }
else
    k=k-1;
n=n/10;
}
cout<<nr;
}
    
```

Varianta PASCAL

```

program bac;
var n,k,nr,p: longint;
begin
nr:=0;
p :=1;
readln(n,k);
while (n<>0) and (k<>0) do
begin
if n mod 2=0 then
begin
nr:=nr+n mod 10*p;
p:=p*10;
end
else
k:=k-1;
n:=n div 10;
end;
write(nr);
end.
    
```

SUBIECTUL II

1. b pentru intensiv și c pentru neintensiv

2. c pentru intensiv și a pentru neintensiv

3.abefgh 6

4.Varianta C/C++

Pentru intensiv

Trebuie initializat cu primul element al listei

```

S=p->info;
while(p)
{p=p->urm;
s=s+p->info;}
    
```

Varianta PASCAL

```

S:=p^.info;
while p^.urm<>NIL do
begin
p:=p^.urm;
s=s+p^.info;
end;
    
```

```
cout<<s;/ printf("%d",s);
```

4. Pentru neintensiv

Varianta C/C++

```
#include<iostream.h>
void main( )
{int a[20][20],n,p,i,j,k=1;
do{
cout<< "n=";
cin>>n;
cout<< "p=";
cin>>p;
}while(2>n || n>20 || 1>p || p>20);
for(i=1;i<=n;i++)
for(j=1;j<=p;j++)
    {a[i][j]=k*k;
k=k+2;
}
    for(i=1;i<=n;i++)
    {for(j=1;j<=p;j++)
    cout<<a[i][j]<<" ";
    cout<<endl;
    }
}
```

```
end;
write(s);
```

Varianta PASCAL

```
PROGRAM BAC;
var
a: array [1..20,1..20] of integer;
n,p,i,j,k:integer;

begin
k:=1;
repeat
begin
write('n=');
readln(n);
write('p=');
readln(p);
end;
until ((2<n) and (n<20) and (p>1)
and (p<20));
for i:=1 to n do
for j:=1 to p do
begin
a[i][j]:=k*k;
k:=k+2;
end;
for i:=1 to n do
begin
for j:=1 to p do
write(a[i,j], ' ');
writeLn;
end;
end.
```

SUBIECTUL III

1.c pentru intensiv a pentru neintensiv

2.. 2

1 2 3

pentru neintensiv5 (105,123,231,321,501)

3.Varianta C/C++

Pentru intensiv

```
int sub(int unsigned n)
{int s=0,x;
for(int i=1;i<=n;i++)
{
cout<<"x=";
cin>>x;

if(sqrt(x)==int(sqrt(x)))
s=s+x;
}
return s;
}
```

Varianta PASCAL

Pentru intensiv

```
function sub(n:integer):integer;
var s,x,i:integer;
begin
s:=0;
for i:=1 to n do
begin
write('x=');
readln(x);
if
sqrt(x)=int(sqrt(x)) then
s:=s+x;
end;
sub:=s;
end;
```

3. Varianta C/C++

Pentru neintensiv

```

:
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
FILE*f=fopen("BAC.TXT","w");
void main()
{ int i,j,a[100][100],n,k=0;
do {
cout<<"n=";
cin>>n;}while(n>50);
for( i=1;i<=n;i++)
{for(j=1,k=0;j<=n-i+1;j++)
{
a[i][j]=k;
k=k+2;
}}

for(i=1;i<=n;i++)
{for(j=1;j<=n-i+1;j++)

fprintf(f,"%d ",a[i][j]);
fprintf(f,"\n");
}
}.
    
```

4. a) Varianta C/C++

Pentru intensiv

```

#include<iostream.h>
#include<stdio.h>
FILE*f=fopen("BAC.TXT","w");
int m,n;
int a[100],b[100];int c[100];

int min(int x,int y)
{if(x>y)
{c[1]=y;
return 0;
}
else
{ c[1]=x;
return 1 ;
}
}

void citire()
{
int i;
cout<<"n=";
cin>>n;
cout<< "m=";

cin>>m;
for(i=1;i<=n;i++)
{cout<<"a["<<i<<"]=";
cin>>a[i];}
    
```

Varianta PASCAL

```

program bac;
var f:text;
i,j,n,k:integer;
a: array [1..100,1..100] of
integer;
begin
assign(f,'BAC.TXT');
rewrite(f);
k:=0;
repeat
write('n=');
readln(n);
until n<50;
for i:=1 to n do
begin
k:=0;
for j:=1 to n-i+1 do
begin
a[i][j]:=k;
k:=k+2;
end;
end;
for i:=1 to n do
begin
for j:=1 to n-i+1 do
write(f,' ',a[i,j]);
writeln(f);
end; close(f);
end.
    
```

Varianta PASCAL

```

program bac ;
var
m,n,i,j,k:integer;
a,b,c:array[1..100] of integer;
f:text;
function min(x,
y:integer):integer;
begin
if x>y then
begin
c[1]:=y;
min:=0;
end
else
begin
c[1]:=x;
min:=1 ;
end;
end;

procedure citire;
var i:integer;
begin
readln(n,m);
for i:=1 to n do
begin
write('a[' ,i, ']=');
    
```

```

    for(i=1;i<=m;i++)
        {cout<<"b["<<i<<"]=";
          cin>>b[i];}
    }
void main()
{
    citire();
    int i,j,k=1;
    if (min(a[1],b[1]))
    {
        i=2;
        j=1;
    }
    else
    { i=1;
      j=2;
    }

    while(i<=n || j<=m)
    {
        if(a[i]<b[j])
        if((c[k]%2!=a[i]%2) && c[k]<a[i])
        c[++k]=a[i++];
        else
        i++;
        else
        if((c[k]%2!=b[j]%2) && c[k]<b[j])
        c[++k]=b[j++];
        else
        j++;
    }

    for(i=1;i<=k;i++)
    fprintf(f,"%d ",c[i]);
}

```

4. a) Varianta C/C++ Pentru neintensiv

```

int p1( long int n ) ;
int p2( long int n);

```

```

    read(a[i]);
    end;
    for i:=1 to m do
    begin
        write('b[' ,i,']=');
        read(b[i]);
    end;
    end;
begin
    assign(f, 'BAC.TXT');
    rewrite(f);
    citire;
    k:=1;
    if min(a[1],b[1])=1 then
        begin
            i:=2;
            j:=1;
        end
    else
        begin
            i:=1;
            j:=2;
        end;

    while (i<=n ) or (j<=m) do
    begin
        if a[i]<b[j] then
            if ((c[k] mod 2) <> (a[i] mod 2)
            )and (c[k]<a[i]) then
                begin
                    k:=k+1;
                    c[k]:=a[i];
                    inc(i);
                end
            else
                inc(i)
            else
                if ((c[k] mod 2 )<>(b[j] mod 2))
                and ( c[k]<b[j]) then
                    begin
                        inc(k);
                        c[k]:=b[j];
                        inc(j);
                    end
                else
                    inc(j);
            end;

        for i:=1 to k do
        write(f,c[i], ' ');
        close(f);
    end.

```

Varianta PASCAL

```

function p1(n:longint):integer;
function p2(n:longint):integer;

```

4. b) Pentru intensiv

Eficiența metodei utilizate va ține seama de faptul că cei doi vectori introduși sunt ordonați și au componente de paritate diferită. astfel vom lua un element dintr-un tablou și celălalt element din cel de-a doilea tablou cât timp $a[i] < b[j]$. În cazul în care nu mai este asigurată continuitatea se caută primul element de paritate diferită decât ultimul element pus în tabloul c ținând cont ca acesta se alege din a sau b. Pentru eficiență se va căuta de la ultima poziție în care s-a făcut ultima alegere.

4. b) Pentru neintensiv

Varianta C/C++

```
Pentru neintensiv
#include<iostream.h>
int p1(int long n)
{int s=0;
    while(n!=0)
    { s=s+n%10;
      n=n/10;
    }
return s;
}

int p2(int long n)
{
return n/10;
}

void main ( )
{long int n,nr=0;

do
cin>>n;
}while(n==0);
    while(n!=0)
        {if (p1 (n)==p1 (p2 (n)))
            nr++;
            n=p2 (n);
        }

cout<<nr;
}
```

Varianta PASCAL

```
program bac;
var
n,nr:longint;

function p1(n:longint):integer;
var s:integer;
begin
s:=0;
    while n<>0 do
    begin
s:=s+n mod 10;
n:=n div 10;
end ;
p1:=s;
end;

function p2(n:longint):integer;
begin
p2:=n div 10;
end;

begin
nr:=0;
repeat
write('n=');
readln(n);
until n<>0;
    while n<>0 do
    begin
if p1 (n)=p1 (p2 (n))
then
inc(nr);
n:=p2 (n);
end;
write(nr);
end.
```

Varianta <9>**SUBIECTUL I**

1. b

2. a) 1

2. b) orice număr care conține macar un 0. Exemplu 102,10, 100,

2. c)

Citește n (număr natural)

```

s ← 10
repetă
    dacă n%10<s atunci
        s ← n%10
    altfel
        s ← -1
    n=[n%10]
până când (n=0)
scrie s
    
```

2.d) Varianta C/C++

```

#include <iostream.h>
void main()
{
    int n, s;
    cout<<"Dati n:"; cin>>n;
    s=10;
    while (n> 0)
    {
        if(n%10<s)
            s=n%10;
        else
            s=-1;
        n=n/10;
    }
    cout<<s;
}
    
```

Varianta PASCAL

```

program bac;
var n,s:integer;
begin
    write('Dati n:'); readln(n);
    s:=10;
    while n> 0 do
        begin
            if n mod 10<s then
                s:=n mod 10
            else
                s:=-1;
            n:=n div 10;
        end;
    write(s);
end.
    
```

SUBIECTUL II

1. a pentru intensiv și c pentru neintensiv

2. a pentru intensiv și c pentru neintensiv

3.4 pentru intensiv și abcdea pentru neintensiv dacă în loc de a[] se consideră s[]

Pentru neintensiv în c/c++ greșeală de scriere, variabila a nu este definită

4. abcd123efg pentru intensiv și 5 pentru neintensiv

5.Varianta C/C++

```

#include <iostream.h>
#include <conio.h>
void main()
{ clrscr();
int n, i, j, a[24][24];
cout<<"n= ";cin>>n;
cout<<"Dati matricea..\n";
for(i=1; i<=n; i++)
for(j=1; j<=n; j++)
cin>>a[i][j];
for (i=1; i<=n; i++)
cout<<a[1][i]<<' ';
for (i=1; i<=n; i++)
    cout<<a[i+1][n]<<' ';

for(i=n; i>=1; i--)
for (i=n-1; i>1; i--)
cout<<a[i][1]<<' ';
}
    
```

Varianta PASCAL

```

PROGRAM BAC;
var a:array[1..24,1..24] of
integer;
n, i, j:integer;
begin
    write('n=');
    read(n);
    writeln('Dati matricea...');
    for i:=1 to n do
        for j:=1 to n do
            readln(a[i,j]);
    for i:=1 to n do
        write(a[1,i],' ');
    for i:=1 to n-1 do
        write(a[i+1,n]<down to 1) do
            write(a[n,i],' ');
    for i:=n-1 downto 2 do
    
```

```
} write(a[i,1], ' ');
end.
```

SUBIECTUL III

1. b pentru intensiv și a pentru neintensiv

2. 1 pentru intensiv și 156 pentru neintensiv

3. Varianta C/C++

Pentru intensiv

```
#include<fstream.h>
#include<conio.h>
int v[100];
void sub(int n,int k)
{
int i,c=1;
for (i=1;i<=n;i++)
{
v[c]=v[c]+i*k;
c++; }
cout<<"\n";

for(i=n;i>=1;i--)
cout<<v[i]<<" ";
}
```

3.a) Varianta C/C++

Pentru neintensiv

```
void s2(int &a, int &b);
```

3.b) Varianta C/C++

Pentru neintensiv

```
void s1(int a[100], int &p, int &q)
{
int i, ep=-1, ei=-1;
for (i=p; i<=q; i++)
{
if(ep==-1&& a[i]%2==0) ep=i;
if(a[i]%2==1) ei=i;
}
p=ep; q=ei;
}
```

3.c) Varianta C/C++

Pentru neintensiv

```
#include <fstream.h>
#include <conio.h>
ofstream f("bac.txt");
void s1(int a[100], int &p, int &q)
```

Varianta PASCAL

Pentru intensiv

```
var v :array[1..100] of integer;
procedure sub( n, k:integer);
var
i,c:integer;

begin
c:=1;
for i:=1 to n do
begin
v[c]:=v[c]+i*k;
inc(c);
end;
writeln;

for i:=n downto 1 do
write (v[i], ' ');
end;
```

Varianta PASCAL

Pentru neintensiv

```
procedure s2 (var a:integer; var
b:integer);
```

Varianta PASCAL

Pentru neintensiv

```
type tablou=array[1..100] of
integer;
procedure s1(a:tablou; var
p:integer; var q:integer);
var i,ep,ei:integer;
begin
ep:=-1; ei:=-1;
for i:=p to q do
begin

if (ep=-1) and (a[i] mod 2=0)
then
ep:=i;
if (a[i] mod 2=1) then
ei:=i;
end;
p:=ep; q:=ei;
end;
```

Varianta PASCAL

Pentru neintensiv

```
program bac;
type tablou=array[1..100] of
integer;
var a:tablou;
```

```

{
int i, ep=-1, ei=-1;
for (i=p; i<=q; i++)
    {
        if(ep==-1&& a[i]%2==0) ep=i;
        if(a[i]%2==1) ei=i;
    }
p=ep; q=ei;
}

void afis_v(int v[100], int a, int
b)
{
for(int i=a; i<=b;i++)
    f<<v[i]<<' ';
f<<endl;
}
void s2 (int &a, int &b)

{ int aux;
a=a;
a=b;
b=aux;
}
void main()
{ int n,a[100], i_par, i_imp;
clrscr();
cout<<"n= ";cin>>n;
cout<<"\nDati vectorul!\n";
for (int i=1; i<=n; i++)
    cin>>a[i];
i_par=1;
i_imp=n;
while (i_par!=-1||i_imp!=-1)
    {
        s1(a, i_par, i_imp);
        if(i_par<i_imp)
            s2(a[i_par], a[i_imp]);
    }
cout<<endl;
afis_v(a,1,n);
f.close();
getch();
}

```

4. a) Pentru intensiv

Varianta C/C++

```

#include<stdio.h>
#include<conio.h>

```

```

i,n,i_par,i_imp:integer;
procedure s1(a:tablou; var
p:integer; var q:integer);
var i,ep,ei:integer;
begin
ep:=-1; ei:=-1;
for i:=p to q do
    begin
        if (ep=-1) and (a[i] mod 2=0)
then
            ep:=i;
            if (a[i] mod 2=1) then
                ei:=i;
            end;
p:=ep; q:=ei;
end;

procedure afis_v (v:tablou;
a:integer; b:integer);
var i:integer;
begin
for i:=a to b do
    write(f,v[i],' ');
writeln(f);
end;

procedure s2 (var a:integer; var
b:integer);
var aux:integer;
begin
aux:=a;
a:=b;
b:=aux;
end;

begin
assign(f,'bac.txt');
rewrite(f);
write('n= ');readln(n);
writeln('Dati vectorul!');
for i:=1 to n do
    readln(a[i]);
i_par:=1;
i_imp:=n;
while (i_par<>-1) or (i_imp<>-1) do
    begin
        s1(a, i_par, i_imp);
        if i_par<i_imp then
            s2(a[i_par], a[i_imp]);
        end;
writeln;
afis_v(a,1,n);
close(f);
end.

```

Pentru intensiv

Varianta PASCAL

```

program bac;
var v:array[1..1000]of integer;

```



```

int v[1000];
FILE *f=fopen("BAC.TXT","r");
void main ()
{clrscr();
int i,x,c=0;

while(!feof(f))
{fscanf(f,"%d",&x);
if(x/100<=9 && x/100 >=1)
v[x]=1;
}
for(i=999;i>=100;i--)
{if( v[i]==0)
{printf("%d ",i);
c++;
}
}
if(c==2)

break;}

```

```

i,x,c:integer;
f:text;
begin
assign(f,'BAC.TXT');
reset(f);
c:=0;

while not eof(f) do
begin
read(f,x);
if (x div 100<=9 ) and (x div 100
>=1) then
v[x]:=1;
end;

for i:=999 downto 100 do
begin
if v[i]=0 then

begin
write(i,' ');
inc(c);
end;
if c=2 then
break;
end
end.

```

Parcurgem o singurădat ăfi șierul de intrare ,care conține numerele.Marchez în vectorul v numerele de trei cifre existente în fișier. Parcurgând descrescător verific care nu se regăsesc în vector.

Varianta <10>

SUBIECTUL I

1. a

2.

a) 24

b) n=233456, k=4

n=237458, k=3

c)

citește n,k (numere naturale nenule)

nr ← 0

p ← 1

dacă n ≠ 0 și k ≠ 0

┌ execută

┌ dacă n%2 ≠ 0 atunci

┌ nr ← nr + [n/10] % 10 * p

┌ p ← p * 10

┌ altfel

┌ k ← k - 1

└ ──

┌ n ← n / 10

└ cât timp n ≠ 0 și k ≠ 0

scrie nr

d) Varianta C/C++

```
#include <iostream.h>
unsigned long n,k,p,nr;

void main()
{cout<<"Dati n si k:"; cin>>n>>k;
nr=0; p=1;
while(n!=0 && k!=0)
{if(n%2!=0)          {nr=nr+n%10*p;
p=p*10;} else k=k-1;
n=n/10;}
cout<<nr;}
```

Varianta PASCAL

```
var n,k,p,nr:longint;
begin write('Dati n si k:'); readln(n,k);

while((n<>0)and(k<>0))do
begin if(n mod 2<>0)then begin
nr:=nr+(n mod 10)*p; p:=p*10 end
else k:=k-1;

n:=n div 10
end;
write(nr)
end.
```

SUBIECTUL II

1. a

Intensiv

2.d (se afisează conținutul nodurilor, începând de la al treilea nod, din două în două noduri, până la sfârșitul listei)
 3.3081 (Un arbore cu n noduri are n-1 muchii, așadar graful nostru cu 80 de noduri trebuie să aibă 79 de

muchii, deci trebuie eliminate 3160-

79=3081 deci trebuie eliminate 3160-79=3081) **4.abacde** (literele consecutive o singură dată)

Neintensiv

2.c (șterge caracterele ce se repetă, cu reveniri la caracterul precedent după fiecare ștergere)

3.3 (Întrucât două din cele 6 noduri au gradul 0, rezultă că celelalte 4 noduri sunt legate prin cele 6 muchii, lucru

posibil doar dacă fiecare din cele 4 noduri are gradul 3) **4.3481** (Un arbore cu n noduri are n-1 muchii, așadar graful nostru cu 80 de noduri trebuie să aibă 79 de muchii, deci trebuie eliminate 3560-79=3481).

5.

(neintensiv și intensiv). Numerele pătrate perfecte care trebuie introduse sunt $(2 \cdot 0)^2, (2 \cdot 1)^2, \dots, [2 \cdot (n \cdot p - 1)]^2$. Considerăm tabloul începând cu linia 0 și coloana 0. Elementul general din linia i și coloana j al tabloului este elementul numărul $i \cdot p + j + 1$ din șirul pătratelor (căci avem i linii – de la 0 la i-1 – de câte p elemente, iar elementul este în poziția j+1 în linia i), adică este elementul $[2 \cdot (i \cdot p + j)]^2$.

Varianta C/C++

Varianta PASCAL

```
#include <iostream.h>
#include <math.h>
int i,j,n,p; unsigned long a[20][20];
void main()
{do{cout<<"Dati n[2-20]:"; cin>>n;}
while(n<2 || n>20);
do{cout<<"Dati p[1-20]:"; cin>>p;}
while(p<1 || p>20);
for(i=0;i<n;i++)
for(j=0;j<p;j++)
a[i][j]=pow(2*(i*p+j),2);
for(i=0;i<n;i++)
{for(j=0;j<p;j++) cout<<a[i][j]<<'
'; cout<<endl;}}
```

```
var i,j,n,p:integer; a:array[0..20,0..20]of
longint;
begin repeat write('Dati n[2-20]:');
readln(n) until(n>1)and(n<=20);
repeat write('Dati p[1-20]:'); readln(p)
until(n>0)and(n<=20);
for i:=0 to n-1 do
for j:=0 to p-1 do
a[i][j]:=sqr(2*(i*p+j));
for i:=0 to n-1 do
begin for j:=0 to p-1 do write(a[i][j],' ');
writeln end;
readln
end.
```

SUBIECTUL III

1. a pentru **intensiv** și b pentru **neintensiv**

$f(3,1)=2+f(2,2)=2+3+f(1,3)=5+4+f(0,4)=9+0=9$.

2. 5 (dintre cifrele 5,6,7,8 și 9, care apar în această ordine trebuie alese 4, deci una lipsește. Cum sunt 5 cifre există 5 posibilități ca o cifră să lipsească. Numerele sunt 45678, 45679, 45689, 45789, 46789).

3.

Varianta C/C++

Pentru **intensiv**
Pornim cu $a=n-1$ și cât timp a nu e prim îl decrementăm cu 1. Pornim cu $b=a-1$ și cât timp nu e prim îl decrementăm cu 1.

```
void sub(unsigned long n,
unsigned long &a, unsigned long
&b)
{unsigned long i;int prim;
a=n-1;
do{prim=1;
for(i=2;i*i<=a;i++)
if(a%i==0) prim=0;
if(!prim) a--;} while(!prim);
b=a-1;
do{prim=1;
for(i=2;i*i<=b;i++)
if(b%i==0) prim=0;
if(!prim) b--;} while(!prim);}
```

Varianta PASCAL

```
procedure sub(n:integer;var a,b:longint);
var i:integer; prim:boolean;
begin a:=n-1;
repeat prim:=true;
for i:=2 to trunc(sqrt(a)) do
if (a mod i=0) then prim:=false;
if(prim=false) then dec(a) until
prim=true;
b:=a-1;
repeat prim:=true;
for i:=2 to trunc(sqrt(b)) do
if(b mod i=0) then prim:=false;
if(prim=false) then dec(b) until
prim=true;
end;
```

Pentru neintensiv

Memorăm în vectorul s sumele cerute, în care $s[i]=a[1]+a[2]+\dots+a[i]$. Avem: $s[i]=(a[1]+a[2]+\dots+a[i-1])+a[i]=s[i-1]+a[i]$, pentru orice i de la 1 la n . $s[0]=0$.

```
#include <iostream.h>
#include <conio.h>
int i,n; unsigned long
```

```
Varianta Pascal  
var i:integer; a,s:array[0..100]of
longint;
begin repeat write('Dati n[1-100]:');
```

```

a[101],s[101];
void main()
{do{cout<<"Dati n[1-100]:";
cin>>n;}
while(n<1 || n>100);
s[0]=0;
for(i=1;i<=n;i++)
{do{cout<<"a["<<i<<"]=";
cin>>a[i];} while(a[i]>9999);
s[i]=s[i-1]+a[i];}
for(i=1;i<=n;i++) cout<<s[n+1-
i]<<endl;
getch();}

```

4. a)

Varianta C/C++

Pentru intensiv

```

#include <fstream.h>
ifstream f("produse.txt");
unsigned long x[10000]; int
i,j,k;
void main()
{while(f>>i>>j>>k) x[i]+=j*k;
f.close();
for(i=0;i<=9999;i++)
if(x[i]) cout<<i<<'
'<<x[i]<<endl;}

```

Pentru neintensiv

```

#include <fstream.h>
#include <conio.h>
unsigned long i,n,p,x; ofstream
f("BAC.txt");
void main()
{do{cout<<"Dati n[1-1000000]:";
cin>>n;} while(n<1 ||
n>1000000);
p=1; cout<<"Dati numerele:";
for(i=1;i<=n;i++)
{cin>>x; while(p<x) p*=2;
f<<p<<' ';}
f.close();}

```

b) Pentru intensiv

În vectorul x păstrăm cantitățile totale: $x[i]$ este suma totală obținută pentru produsul cu codul i . Pentru fiecare tripletă i,j,k citit ă din fișier incrementăm pe $x[i]$ cu $j*k$, fără a mai căuta prin vectori sau sorta vectori. Astfel, sunt necesare doar n adunări și n înmulțiri, unde n e nr. de înregistrări din fișier.

```

readln(n) until (n>0) and (n<=100);
s[0]:=0;
for i:=1 to n do
begin repeat write('a[' ,i,']=');
readln(a[i]) until (a[i]<=9999);
s[i]:=s[i-1]+a[i] end;
for i:=1 to n do writeln(s[n+1-i]);
readln
end.

```

Varianta PASCAL

```

var f:text; x:array[0..10000]of longint;
i,j,k:integer;
begin assign(f,'produse.txt'); reset(f);
while not (eof(f)) do
begin readln(f,i,j,k); x[i]:=x[i]+j*k
end;
for i:=0 to 9999 do
if (x[i]>0) then writeln(i,' ',x[i]);
close(f)
end.

```

```

var i,n,p,x:longint; f:text;
begin assign(f,'BAC.txt'); rewrite(f);
repeat write('Dati n[1-1000000]:');
readln(n) until ((n>0) and (n<1000000));
p:=1; write('Dati numerele:');
for i:=1 to n do
begin read(x); while (p<x) do p:=p*2;
write(f,p,' ') end;
close(f)
end.

```

b) Pentru neintensiv

Memorăm în p puterea lui 2 căutată. Inițial, $p=1$. Pentru fiecare număr citit îl dublăm pe p (dacă e mai mic decât numărul) până devine cel puțin egal cu numărul citit și scriem pe p . Astfel, nu reluăm cu $p=1$ pentru fiecare număr, ci continuăm de la pasul precedent, dacă e cazul. Fie $p=2^k$, deci $2^{k-1} < x \leq 2^k$ (unde x e ultimul număr din șir), căci k e minim.

Logaritmăm în baza 2 și obținem $k-1 < \log_2 x \leq k$, deci k (numărul de înmulțiri cu 2 efectuate) este $\log_2 x$ dacă x e putere a lui 2 sau $1 + [\log_2 x]$,

| dacăx nu e putere a lui 2.

Varianta <11>

SUBIECTUL I

- 1. c
- 2.

a) a)15 (se afișează produsul ultimelor p cifre ale lui n, unde n e minimul dintre k și nr. de cifre ale lui n)

b) 1 (sau orice alt număr care nu are printre ultimele 4 cifre 3,5,7 și 9).

c) Varianta C/C++

```
#include <iostream.h>
unsigned long n,k,p; int c;
void main()
{cout<<"Dati n si k:"; cin>>n>>k;
p=1;
while(n>0 && k>0)
{c=n%10;
if(c%2==1) p=p*c;
n=n/10; k=k-1;}
cout<<p;}
```

Varianta PASCAL

```
var n,k,p:longint; c:integer;
begin write('Dati n si k:');
readln(n,k);
p:=1;
while((n>0) and(k>0))do
begin c:=n mod 10;
if(c mod 2=1) then p:=p*c;
n:=n div 10; k:=k-1 end;
write(p)
end.
```

d)

citește n, k (numere naturale

```
p←1
dacă k>0
| pentru i=k,1,-1 execută
| | dacă n>0 atunci
| | | c←n%10
| | | dacă c%2=1 atunci
| | | | p←p*c
| | n←[n/10]
| | k←k-1
|
scrie p
```

SUBIECTUL II

- 1. a (drumul este 4,5-1-2,6 și are 4 arce)
- 2. Pentru **intensiv**: d (nr. de grafuri orientate fără bucle cu n noduri este $2^{n(n-1)}$. Pentru n=4 avem $2^{4*3}=2^{12}=4096$).

Pentru **neintensiv**,d

- 3.

11 brcdbr (se afișează lungimea inițială a lui s, spațiu și apoi șirul s fără vocala a.

- 4.

Pentru intensiv

p->nr+p->urm->nr+p->urm->urm->nr (pentru C++), respectiv p^.nr+p^.urm^.nr+p^.urm^.urm^.nr (pentru Pascal)

Pentru neintensiv

64 (nr. de grafuri neorientate distincte cu vârfuri e $2^{n(n-1)/2}$. Pentru n=4 avem

$$2^{4 \cdot 3/2} = 2^6 = 64.$$

5.

Pentru intensiv

Varianta C/C++

```
#include <iostream.h>
int a[10][10], i, j, m, max, min, n;
void main()
{do{cout<<"Dati m[1-10]:"; cin>>m;}
while(m<1 || m>10);
do{cout<<"Dati n[1-10]:"; cin>>n;}
while(n<1 || n>10);
for(i=0; i<m; i++)
for(j=0; j<n; j++)
{cout<<"a["<<i<<"] ["<<j<<"]=";
cin>>a[i][j];}
max=0; //maximul dintre minimele
liniilor

for(i=0; i<m; i++)
{min=a[i][0]; //minimul liniei i
for(j=1; j<n; j++)
if(a[i][j]<min) min=a[i][j];
if(max<min) max=min;}
cout<<"Maximul minimelor liniilor
este "<<max;}
```

Pentru neintensiv

```
#include <iostream.h>
int a[10][10], i, j, m, min, n;
void main()
{do{cout<<"Dati m[1-10]:"; cin>>m;}
while(m<1 || m>10);
do{cout<<"Dati n[1-10]:"; cin>>n;}
while(n<1 || n>10);
for(i=0; i<m; i++)
for(j=0; j<n; j++)
{cout<<"a["<<i<<"] ["<<j<<"]=";
cin>>a[i][j];}
for(i=0; i<m; i++)
{min=a[i][0];
for(j=1; j<n; j++)
if(a[i][j]<min) min=a[i][j];
cout<<min<<' ' ;}}
```

Varianta PASCAL

```
var a:array[0..10,0..10]of integer;
i, j, m, max, min, n:integer;
begin repeat write('Dati m[1-10]:');
readln(m) until (m in[1..10]);
repeat write('Dati n[1-10]:');
readln(n) until(n in[1..10]);
for i:=0 to m-1 do
for j:=0 to n-1 do
begin write('a[' , i, ']' [' , j, ']=' );
readln(a[i, j])end;
max:=0; {maximul dintre minimele
liniilor}
for i:=0 to m-1 do
begin min:=a[i][0]; {minimul liniei
i}
for j:=1 to n-1 do
if(a[i, j]<min) then min:=a[i, j];
if(max<min) then max:=min end;
writeln('Maximul minimelor
liniilor este ', max)
end.
var a:array[0..10,0..10]of integer;
i, j, m, max, min, n:integer;
begin repeat write('Dati m[1-10]:');
readln(m)
until (m in[1..10]);
repeat write('Dati n[1-10]:');
readln(n)
until(n in[1..10]);
for i:=0 to m-1 do
for j:=0 to n-1 do
begin write('a[' , i, ']' [' , j, ']=' );
readln(a[i, j])end;
for i:=0 to m-1 do
begin min:=a[i, 0];
for j:=1 to n-1 do
if(a[i, j]<min) then min:=a[i, j];
write(min, ' ')
end.
end.
```

SUBIECTUL III

1. b

2.

Pentru intensiv: 1 (funcția întoarce cea mai mică cifră a lui n, dacă $n \neq 0$, și 9 dacă $n=0$; deci $f(23159)=1$).

Pentru neintensiv: (funcția întoarce suma cifrelor pare ale lui n, deci $f(23169)=8$).

3.

Pentru intensiv

a) Citim în variabila x câte un număr din fișier și în max calculăm maximul cerut. Inițial, max e primul număr citit. Pentru fiecare număr x citit, max devine x dacă

$x > \max$ sau își păstrează vechea valoare în caz contrar, nemaifiind nevoie să comparăm cu toate numerele anterioare. Astfel se evită utilizarea vectorilor, deci se face economie de memorie și timp. Se fac doar $n+1$ citiri din fișier, n comparații și cel mult n atribuiri și folosim doar 2 variabile simple, x și \max .

Varianta C/C++

```
b)
#include <fstream.h>
int i,n; long x,max; ifstream
f("numere.txt");
void main()
{f>>n>>x; cout<<x; max=x;
  for(i=2;i<=n;i++)
  {f>>x; if(x>max) max=x;
  cout<<' '<<max;}
  f.close();}
```

Varianta PASCAL

```
var i,n:integer; x,max:longint; f:text;
begin assign(f,'numere.txt'); reset(f);
  readln(f,n);
  read(f,x); write(x); max:=x;
  for i:=2 to n do
  begin read(f,x); if(x>max)then max:=x;
  write(' ',max) end;
  close(f)
end.
```

Pentru neintensiv

Varianta C/C++

```
void p(long a[100],int k,long
&max)
{int i;
  max=a[1];
  for(i=2;i<=k;i++)
  if(a[i]>max) max=a[i];}
```

Varianta PASCAL

```
type vector=array[0..100]of longint;
procedure p(a:vector; k:integer; var
max:longint);
var i:integer;
begin max:=a[1];
  for i:=2 to k do
  if(a[i]>max)then max:=a[i]
end;
```

Varianta C/C++

```
b)
#include <iostream.h>
int i,n; long
a[100],max,v[100];
void p(long a[100],int k,long
&max)
{int i;
  max=a[1];
  for(i=2;i<=k;i++)
  if(a[i]>max) max=a[i];}
void main()
{for(i=1;i<=n;i++)
  {cout<<"a["<<i<<"]="";
  cin>>a[i]; p(a,i,max);
  v[i]=max;}
  for(i=1;i<=n;i++)
  cout<<v[i]<<' '};}
```

Varianta PASCAL

```
type vector=array[0..100]of longint;
var i,n:integer; a,v:vector; max:longint;
procedure p(a:vector; k:integer; var
max:longint);
var i:integer;
begin max:=a[1];
  for i:=2 to k do
  if(a[i]>max)then max:=a[i]
end;
begin write('n='); readln(n);
  for i:=1 to n do write(a[i], ' ');
  readln(a[i]);
  p(a,i,max); v[i]:=max end;
  for i:=1 to n do write(v[i], ' ')
end.
```

4. a)

Varianta C/C++

Pentru intensiv

```
unsigned long sum(unsigned long
x)
```

Varianta PASCAL

```
function sum(x:longint):longint;
```

b)

Varianta C/C++

Varianta PASCAL

```
#include <iostream.h>
unsigned long i,n,nr,x;
unsigned long sum(unsigned long
x)
{int i,s=0;
  for(i=1;i<=x;i++)
    if(x%i==0) s=s+i;
  return s;}
void main()
{cout<<"n="; cin>>n; nr=0;//nr.
de numere prime
  cout<<"Dati numerele:";
  for(i=1;i<=n;i++)
    {cin>>x; if(sum(x)==x+1)
nr++;}
  cout<<nr;}
```

Pentru neintensiv

Varianta C/C++

```
#include <iostream.h>
unsigned long i,n,nr,x;
int prim(unsigned long x)
{int p; unsigned long d;
  if(x%2==0) return x==2; else
  if(x==1) return 0; else
  {p=1; d=3;
  while(p && d*d<=x)
    if(x%d==0) p=0; else d+=2;
  return p;}}
void main()
{cout<<"n="; cin>>n; nr=0;//nr.
de numere prime
  cout<<"Dati numerele:";
  for(i=1;i<=n;i++)
    {cin>>x; if(prim(x)) nr++;}
  cout<<nr;}
```

```
var i,n,nr,x:longint;
function sum(x:longint):longint;
var i,s:longint;
begin s:=0;
  for i:=1 to x do
    if(x mod i=0) then s:=s+i;
  sum:=s
end;
begin write('n='); readln(n); nr:=0;{nr.
de numere prime}
  write('Dati numerele:');
  for i:=1 to n do
    begin read(x); if(sum(x)=x+1) then
nr:=nr+1 end;
  write(nr)
end.
```

Varianta PASCAL

```
var i,n,nr,x:longint;
function prim(x:longint):boolean;
var p:boolean; d:longint;
begin if(x=2) then prim:=true else
  if((x=1) or (x mod 2=0)) then prim:=false
  else
  begin p:=true; d:=3;
  while((p) and (d*d<=x)) do
    if(x mod d=0) then p:=false else
d:=d+2;
  prim:=p;
  end
end;
begin write('n='); readln(n); nr:=0;{nr.
de numere prime}
  write('Dati numerele:');
  for i:=1 to n do
    begin read(x); if(prim(x)) then nr:=nr+1
end;
  write(nr)
end.
```

Varianta <12>

SUBIECTUL I

1. d

2. a) 17396

b) Din valoarea calculată la punctul a), se observă că algoritmul creează un număr folosind prima cifră fiecărui număr din șirul dat. Pentru a obține numărul 321, șirul dat trebuie să conțină un număr ce începe cu 3, un număr ce începe cu 2, un număr ce începe cu 1 și 0.

Exemplu: 31, 244, 15, 0.

c) Varianta C/C++

```
#include<iostream.h>
#include<stdio.h>

void main()
{int x,y=0;
  cout<<"Dati x=";cin>>x;
  // printf("x=");scanf("%d",&x);
```

c) Varianta PASCAL

```
program bac;
var i,n,a: integer;

begin
  readln(a,n);
  for i:=1 to n do
    if (i mod 2 =0) then a:=a-i*i
```



```
while(x)
  {while(x>9) x/=10;
   y=y*10+x;
   cout<<"Dati x=";cin>>x;
   //printf("x=");scanf("%d",&x);}
  cout<<y; // printf("%d",y); }
```

```
else a:=a+i*i;
write(a);
end.
```

d) Citește x

y ← 0

```
executa
  [
    executa
    [
      x ← [x/10]
      cât timp x > 9
      y ← y*10+x
      citește x
    ]
  ]
cât timp x ≠ 0
Scrie y
```

ȘUBIECTUL II

2. a

3. Pentru intensiv

rădăcina este 2 iar fiii lui 3 sunt 4, 6 și 9

3. Pentru neintensiv

```
struct elev {char nume[31];
             int nota1, nota2, nota3;}
e;
```

4. Secvența afișează lungimea șirului de caractere apoi transformă toate vocalele în majuscule. Valorile afișate sunt: **11 AbcdEfghOld**

5. Varianta C/C++

```
#include<iostream.h>
void main()
{ int a[50][50],n,i,j;
do{cout<<"n=";cin>>n;
   // printf("n=");scanf("%d",&n);
}while(n<=2 ||n>=50);
for(i=1;i<=n;i++)
  {a[i][n-i+1]=0;
   for(j=1;j<=n-i+1;j++) a[i][j]=i;
   for(j=i+1; j<=n;j++) a[j][n-i+1]=i;
  }
for(i=1;i<=n;i++)
{for(j=1;j<=n;j++)

  cout<<a[i][j]<<" ";
  printf("%d", a[i][j]);
  cout<<endl;
}
```

5. Varianta PASCAL

```
var a:array[1..50,1..50] of integer;
     n,i,j:integer;
begin
repeat
write('n=');readln(n);
until(n>2) and (n<50);
for i:=1 to n do
begin
a[i,n-i+1]:=0;
for j:=1 to n-i do a[i,j]:=i;
for j:=i+1 to n do a[j,n-
i+1]:=i;
end;
for i:=1 to n do
begin
for j:=1 to n do write(a[i,j], '
'); writeln;
end;
end.
```

SUBIECTUL III

1. c

2. 2+2+2+3, 2+2+5, 2+7

3. Pentru intensiv

9) Algoritmul care validează dacă un număr este divizibil cu 5 se realizează simplă x și divizibilitatea cu 5 pentru numărul mai mic. Când un șir se termină, algoritmul continuă

pentru numerele din al doilea șir.

Algoritmul este eficient deoarece nu necesită spațiu pentru memorarea, valorile sunt prelucrate printr-o singură parcurgere, la citirea lor.

b) Varianta C/C++

```
#include<fstream.h>
ifstream f("nr1.txt");
ifstream g("nr2.txt");

int diviz(long x)
{ if(x%5==0) return 1;
else return 0;
}

void main()
{long x,y;
 f>>x;
 g>>y;
 while(!f.eof() &&!g.eof())

    {if(x>=y; g>>y;}
    while(x<y&&!f.eof())
        {if(diviz(x))
            cout<<x<<" ";
            f>>x;}
    while(x>y&&!g.eof())
        {if(diviz(y))
            cout<<y<<" ";
            g>>y;}
    }
while(!f.eof())
{f>>x;
if(diviz(x))
    cout<<x<<" ";}
while(!g.eof())
{g>>y;
if(diviz(y))
    cout<<y<<" ";}
cout<<endl;
}
```

b) Varianta PASCAL

```
var x,y:longint;
    f,g:text;
function diviz(x:longint):boolean;
begin
    if x mod 5=0 then diviz:=true
    else diviz:=false;
end;
begin
assign(f,'nr1.txt');reset(f);
assign(g,'nr2.txt');reset(g);
read(f,x); read(g,y);
while not eof(f) and not eof(g) do
begin
    if x=y then
        read(f,x);read(g,y);
    end;
    while (x<y) and (not eof(f))
do
    begin
        if diviz(x) then write(x,' ');
        read(f,x);
    end;
    while (x>y) and (not eof(g))
do
    begin
        if diviz(y) then write(y,' ');
        read(g,y);
    end;
end;
while not eof(f) do
begin
read(f,x);
if diviz(x) then write(x,' ');
end;
while not eof(g) do
begin
read(g,y);
if diviz(y) then write(y,' ');
end;
end.
```

3. a) Pentru neintensiv:

Algoritmul propus presupune memorarea într-un vector a primului șir de numere citit, cel format din m numere. Al doilea șir nu va fi memorat, fiecare valoare nou citită va fi căutată în primul șir, prin metoda căutării binare, deoarece vectorul conține elemente ordonate crescător. Vor fi numărate astfel acele elemente ce apar în ambele șiruri, prin variabila nr . Din numărul total de $n+m$ numere, trebuie scăzute cele dublate. Valoarea afișată este $n+m-$

Ordinul de complexitate este $O(\log_2 n)$

Varianta C/C++

```

b) #include<iostream.h>
long v[100];
int n,m;
int caut(int li,int ls,int x)
{
    int m;
    if(li<=ls)
        {m=(li+ls)/2;
        if(x==v[m]) return 1;
        else
            if(x<v[m]) return caut(li,m-1,x);
            else return caut(m+1,ls,x); }
    else return 0; }
void main()
{int i,nr=0;
long x;
cout<<"m=";cin>>m;
// printf("m=");scanf("%d",&m);
cout<<"n=";cin>>n;
// printf("n=");scanf("%d",&n);
cout<<"Dati primul sir";
//printf("Dati primul sir");
for(i=1;i<=m;i++) cin>>v[i];
// scanf("%l",&v[i]);
cout<<"dati al 2-lea sir:";
//printf("dati al 2-lea sir");
for(i=1;i<=n;i++)
    {cin>>x;// scanf("%l",&x);
    if(caut(1,m,x)==1)nr++;}
cout<<"Sunt "<<n+m-2*nr<<" numere
distincte";
//printf("Sunt %d numere
distincte",n+m-2*nr);
}
    
```

Varianta PASCAL

```

b) var v:array[1..100] of longint;
    n,m,i,nr:integer;
    x:longint;
function
caut(li,ls,x:integer):boolean;
var g:boolean;
    m:integer;
begin
    if li<=ls then
        begin
            m:=(li+ls) div 2;
            if x=v[m] then g:=true
            else
                if x<v[m] then g:=caut(li,m-
1,x)
                else g:=caut(m+1,ls,x);
            end
        end
    else g:=false;
    caut:=g;
end;
begin
nr:=0;
write('m=');readln(m);
write('n=');readln(n);
write('Dati primul sir');
for i:=1 to m do read(v[i]);
write('dati al 2-lea sir:');
for i:=1 to n do
    begin
        read(x);
        if caut(1,m,x) then nr:=nr+1;
    end;
writeln('Sunt ',n+m-2*nr,' numere
distincte');
end.
    
```

4. Pentru intensiv:

Varianta C/C++

```

#include<iostream.h>
int verific(long x)
{int c=x%10; x=x/10;

    while(x)
    {if(x%10!=c) return 0;
    x=x/10;}
    return 1;
}
void main()
{long v[20];
int i,n,nr=0;
do{cout<<"n=";cin>>n;
    }while(n<0||n>20);
for(i=1;i<=n;i++)
    cin>>v[i];
for(i=1;i<=n;i++)
    if(verif(v[i]))
        {nr++;cout<<v[i]<<" ";}
Cout<<nr<<endl;
}
    
```

Varianta PASCAL

```

var v:array[1..20] of longint;
    i,n,nr:integer;
function verific(x:longint):boolean;
var c:integer; ok:boolean;
begin
    ok:=true;
    c:=x mod 10; x:=x div 10;
    while x<>0 do
        begin
            if x mod 10<>c then ok:=false;
            x:=x div 10;
        end;
    verific:=ok;
end;
begin nr:=0;
repeat
write('n=');readln(n);
until (n>0) and (n<=20);
for i:=1 to n do readln(v[i]);
for i:=1 to n do
    if verific(v[i]) then
    
```

	<pre>begin nr:=nr+1; write(v[i], ' '); end; end.</pre>
<p>4. Pentru neintensiv</p> <p>a) int nr(long n, int c) b) #include<iostream.h> int nr(long n, int c) { if(n) if(n%10==c) return 1+nr(n/10,c); else return nr(n/10,c); } void main() { long n; int x=0,i; cout<<"n="; cin>>n; // printf("n=");scanf("%d",&n); for(i=0;i<=9;i++) if(nr(n,i))x++; cout<<x; //printf("%d",x); }</p>	<p>a) function nr(n:longint;c:integer):integer; b) var n:longint; x,i,y:integer; function nr(n:longint;c:integer):integer; var x:integer; begin x:=0; repeat if n mod 10=c then x:=x+1; n:=n div 10; until n=0; end; end; var n,x; begin write('n=');readln(n); for i:=0 to 9 do begin x:=nr(n,i); if x>0 then y:=y+1; end; writeln(y); end.</p>

SUBIECTUL I

Varianta <13>

1. b

2. a) 7

b) b=61 sau b=62

c) Varianta C/C++

```
#include<iostream.h>
void main()
{int a,b,k=0,c,i,n;
cout<<"Dati a=";cin>>a;
// printf("a=");scanf("%d",&a);

cout<<"Dati b=";cin>>b;
// printf("b=");scanf("%d",&b);
for(i=a;i<=b;i++)
{n=i; c=0;
while(n)
{if(n%2==1)c++;
n=n/10;}
if(c>0) k++;
}
cout<<"k="<<k;
// printf("k=%d",k);
}
```

c) Varianta PASCAL

```
var a,b,c,n,k,i:integer;
begin
write('dati a:');
readln(a);

write('dati b:');
readln(b);
k:=0;
for i:=a to b do
begin
n:=i;
c:=0;
while(n>0)do
begin
if (n mod 2=1) then
c:=c+1;
n:=n div 10;
end;
```

```

        if (c>0) then
            k:=k+1;
        end;
        write(' k=' , k);
        readln;
    end.
    
```

d) Citește a, b

```

y←0; i←a
cât timp i≤b executa
    n←i; c←0
    cât timp n>0 executa
        dacă n%2=1 atunci
            c←c+1
        n←[n/10]
    dacă c>0 atunci
        k←k+1
Scrie k
    
```

SUBIECTUL II

1. c
2. b pentru intensiv și b pentru neintensiv
3. Pentru intensiv

Notăm prin $L(i)$ lista de adiacență
 $L(1)=\{2, 3\}$, $L(2)=\{1, 4\}$, $L(3)=\{1, 4\}$
 $L(4)=\{2, 3, 5, 6\}$, $L(5)=\{4, 6\}$ $L(6)=\{4, 5\}$

3. Pentru neintensiv

```

struct medicament {char
denumire[31];
float
pret;} m;
    
```

4. Secvența afișează lungimea șirului de caractere apoi elimină toate vocalele din șir. Valorile afișate sunt: 11 bcd fghd

5. Pentru intensiv:

Varianta C/C++

```

#include<iostream.h>
void main()
{ int a[50][50],n,i,j;
do{cout<<"n=";cin>>n;
// printf("n=");scanf("%d",&n);
}while(n<=2 ||n>=50);
for(i=1;i<=n;i++)
{a[i][n-i+1]=0;
for(j=1;j<n-i+1;j++) a[j][i]=i;
for(j=i+1; j<=n;j++)
a[n-i+1][j]=i;
}
for(i=1;i<=n;i++)
{ for(j=1;j<=n;j++)
cout<<a[i][j]<<" ";
// printf("%d", a[i][j]);
}
}
    
```

Varianta PASCAL

```

var a:array[1..50,1..50] of
integer;
n,i,j:integer;
begin
repeat
write('n=');readln(n);
until(n>2) and (n<50);
for i:=1 to n do
begin
a[i,n-i+1]:=0;
for j:=1 to n-i do a[j,i]:=i;
for j:=i+1 to n do a[n-
i+1,j]:=i;
end;
for i:=1 to n do
begin
    
```

```
    cout<<endl; }
}
```

```
    for j:=1 to n do
write(a[i,j], ' '); writeln;
end;
end.
```

5. Pentru neintensiv:

Varianta C/C++

```
#include<iostream.h>
void main()
{int a[50][50],n,i,j;
do{cout<<"n=";cin>>n;
// printf("n=");scanf("%d",&n);
}while(n<=2 ||n>=50);
for(i=1;i<=n;i++)
for(j=1;j<=n;j++)
if(j<n-i+1)a[i][j]=1;
else if(j==n-i+1) a[i][j]=0;
else a[i][j]=2;
for(i=1;i<=n;i++)
{ for(j=1;j<=n;j++)
cout<<a[i][j]<<" ";
// printf("%d", a[i][j]);
cout<<endl; }
}
```

Varianta PASCAL

```
var a:array[1..50,1..50] of integer;
n,i,j:integer;
begin
repeat
write('n=');readln(n);
until (n>2) and (n<50);
for i:=1 to n do
for j:=1 to n do
if j<n-i+1 then a[i,j]:=1
else
if j=n-i+1 then a[i,j]:=0
else a[i,j]:=2;
for i:=1 to n do
begin
for j:=1 to n do
write(a[i,j], ' ');
writeln;
end;
end.
```

SUBIECTUL III

1. d

3. Pentru intensiv:

b) Varianta C/C++

```
#include<iostream.h>
void main()
{int i=1,j,x=0,n;
cout<<"n=";cin>>n;
// printf("n=");scanf("%d",&n);
while(x<n)
{for(j=i;j>0&& x<n;j--)
x++;
i++;}
cout<<j+1<<" ";
// printf("%d",j+1);
}
```

b) Varianta PASCAL

```
var i,j,x,n:integer;
begin
i:=1; x:=0;
write('n=');read(n);
while x<n do
begin
j:=i;
while (j>0) and (x<n) do
begin
x:=x+1;j:=j-1;
end;
i:=i+1;
end;
write(j+1, ' ');
end.
```

3. a) Pentru neintensiv:

Varianta C/C++

```
#include<iostream.h>
void main(){x=0,n;
cout<<"n=";cin>>n;
// printf("n=");scanf("%d",&n);
```

Varianta PASCAL

```
var i,j,x,n:integer;
begin;x=0;
write('n=');read(n);
while x<n do
```

```
while(x<n)
{for(j=i;j>0&&x<n;j--)
{cout<<j<<" ";
// printf("%d",j);
x++;}
i++;}
}
```

```
begin
j:=i;
while (j>0) and (x<n) do
begin
write(j,' '); x:=x+1;j:=j-1;
end;
i:=i+1;
end;
end.
```

4. Pentru intensiv și neintensiv:

Varianta C/C++

a) doar pentru neintensiv:

```
int p(long n, int c)
{if(n)
{if(n%10==c) return p(n/10,c);
else
return (n%10)+10*p(n/10,c); }
}
```

b)

```
#include<fstream.h>
ifstream f("date.in");
ofstream g("date.out");
int p(long n, int c)
{if(n)
{if(n%10==c)
return p(n/10,c);
else return (n%10)+10*p(n/10,c);}
}
void main()
{int i;long n;
while(!f.eof())
{f>>n;
for(i=1;i<10;i=i+2)
n=p(n,i);
if(n>0) g<<n<<" ";}
}
```

Varianta PASCAL

```
var f,g:text;
n,x:longint; i:integer;
procedure invers(n:longint;var
x:longint);
begin
while n<>0 do
begin
x:=x*10+n mod 10; n:=n div
10;
end;
end;
procedure
p(n:longint;c:integer;var
x:longint);
var nr:integer;
begin
if n<>0 then
if n mod 10=c then p(n div
10,c,x)
else
begin
x:=x*10+n mod 10; p(n div
10,c,x);
end;
end;
begin
assign(f,'bac.in');reset(f);
assign(g,'bac.out');rewrite(g);
while not eof(f) do
begin
read(f,n);
for i:=1 to 10 do
if i mod 2=1 then
begin
x:=0; p(n,i,x); n:=0;
invers(x,n);
end;
if n>0 then writeln(n,' ');
end;
end.
```

Varianta <14>

SUBIECTUL I

1. a

2. a)27596

b) Analizând algoritmul dat, se observă că el determină numărul care se formează utilizând cea mai mare cifră din fiecare număr dat în șir (mai puțin 0 care este introdus pentru a întrerupe citirea). În consecință, pentru a obține **752** trebuie introdus un șir de 4 numere din care ultimul 0 și primele 3 trebuie să aibă cifre maxime pe 7, 5 și respectiv 2. De exemplu, un set de date de intrare ar putea fi: 127, 253, 12, 0.

c) Varianta C/C++

```
#include<iostream.h>
#include<stdio.h>
void main()
{
    unsigned int x,n,y,c;
    scanf("%d",&x); //cin>>x;
    n=0;
    while(x!=0)
        { y=x;c=0;
          while(y>0)
              {if(y%10>c)
                  c=y%10;
                y=y/10;}
            n=n*10+c;
            scanf("%d",&x); //cin>>x;
        }
    printf("%d",n); //cout<<n;
}
```

Varianta PASCAL

```
var x,y,n,c:integer;
begin
    readln(x);
    n:=0;
    while x<>0 do
        begin
            y:=x; c:=0;
            while y>0 do
                begin
                    if y mod 10>c then
                        c:=y mod 10;
                    y:=y div 10;
                end;
            n:=n*10+c;
            readln(x);
        end;
    write(n);
end.
```

d) Deoarece introducerea, la prima citire, a valorii 0 pentru x nu influențează cu nimic rezultatul, nu mai este necesară o verificare anterioară a structurilor repetitive și algoritmul devine:

```
citește x (număr natural)
n←0
repetă
    y←x; c←0
    repetă
        dacă y%10>c atunci
            c←y%10
        y←[y/10]
    până când y=0
    n←n*10+c
    citește x
până când x=0
scrie n
```

SUBIECTUL II

1. b

2. c

3. Varianta C/C++

pentru intensiv se va afișa 9 b d u e d
iar pentru neintensiv 9 2

Varianta PASCAL

pentru intensiv se va afișa
9
b d u e d
iar pentru neintensiv
9 2

4. Etichetele frunzelor arborelui sunt: 2, 3, 4, 6

5. **Pentru intensiv** Se parcurge matricea pe coloane și fiecare coloană (de la 1 la numărul de coloane). După completarea unei coloane cu

de cifre a lui n) se completează cu ultima cifră a numărului n . După completarea unei coloane cu ultima cifră a numărului n , acesta își va schimba valoarea în $n=n/10$. Numărul de linii a matricei, respectiv cel de coloane se determină în momentul validării datelor de intrare, pentru care trebuie să determinăm numărul de cifre.

ultima cifră a numărului n , acesta își va schimba valoarea în $n=n/10$.

Varianta C/C++

Program intensiv

```
#include<iostream.h>
#include<stdio.h>
void main()
{ unsigned a[9][9],i,j,nc;
long n,m;
do{scanf("%D",&n);//cin>>n;
nc=0;m=n;
while(m!=0){m=m/10;nc++;}
}while(nc>8 || n<=0);
for(j=1;j<=nc;j++)
{for(i=1;i<=nc;i++)
a[i][j]=n%10;
n=n/10;}
for(i=1;i<=nc;i++)
{for(j=1;j<=nc;j++)
printf("%d ",a[i][j]);
//cout<<a[i][j]<<" ";
printf("\n");//cout<<endl;
}}
```

Varianta PASCAL

Program intensiv

```
type mat=array[1..8,1..8] of
integer;
var a:mat;
i,j,nc:integer;
m,n:longint;
begin
repeat
readln(n); nc:=0;m:=n;
while(m<>0) do
begin
m:=m div 10; nc:=nc+1;
end;
until (nc<=8) and (n>0);
for j:=1 to nc do
begin
for i:=1 to nc do
a[i,j]:=n mod 10;
n:=n div 10;
end;
end;
for i:=1 to nc do
begin
for j:=1 to nc do
write(a[i,j], ' ');
writeln;
end; end.
```

Varianta C/C++

Pentru neintensiv

```
#include<iostream.h>
#include<stdio.h>
void main()
{ unsigned a[5][5],n,i,j;
do{scanf("%d",&n);//cin>>n;
}while(n<1000 || n>9999);
for(j=1;j<=4;j++)
{for(i=1;i<=4;i++)
a[i][j]=n%10;
n=n/10;}
for(i=1;i<=4;i++)
{for(j=1;j<=4;j++)
printf("%d ",a[i][j]);
//cout<<a[i][j]<<" ";
printf("\n");//cout<<endl;
}}
```

Varianta PASCAL

Pentru neintensiv

```
type mat=array[1..4,1..4] of
integer;
var a:mat;
i,j:integer;
begin
repeat
readln(n);
until (n>=1000) and (n<=9999);
for j:=1 to 4 do
begin
for i:=1 to 4 do
a[i,j]:=n mod 10;
n:=n div 10; end;
for i:=1 to 4 do
begin
for j:=1 to 4 do
write(a[i,j], ' ');
writeln;
end; end.
```

SUBIECTUL III**1. b****2. 8 pentru intensiv și 72 pentru neintensiv****3. Pentru intensiv**

a) Folosim un vector c cu 10 elemente, unul corespunzător fiecărei cifre. $C[i]$ va reprezenta numărul de citiri a cifrei i . În momentul citirii nu vom reține cele n numere ci vom forma vectorul c . Vectorul c este inițial 0 și la fiecare citire adăugăm 1 elementului corespunzător cifrei citite. Pentru a afișa cifrele citite în ordine crescătoare vom parcurge vectorul c și vom afișa fiecare cifră ($i=0..9$) de atâtea ori de câte ori a fost citit ă, adică de $c[i]$ ori. Metoda este mai eficientă din punct de vedere a spațiului de memorie utilizat (pentru ca nu reținem toate cele n cifre) și din punct de vedere a timpului de execuție (pentru că nu este necesară sortarea celor n numere).

b) Varianta C/C++

```
#include<iostream.h>
int c[10];
void main()
{
    int n,cif;
    do{cin>>n;}while(n>500);
    for(int i=1;i<=n;i++)
        {    cin>>cif;
          c[cif]++;}
    for(i=0;i<=9;i++)
        for(int
j=1;j<=c[i];j++)
            cout<<i<<" ";
}
}
```

b) Varianta PASCAL

```
var c:array[0..10] of integer;
    cif,i,j,n:integer;
begin
    repeat
        read(n);
    until n<=500;
    for i:=1 to n do
        begin
            read(cif);
            c[cif]:=c[cif]+1;
        end;
    for i:=0 to 9 do
        for j:=1 to c[i] do
            write(i,' ');
        end;
end.
```

Pentru neintensiv:

Vom citi toate datele din fișierul de intrare (până la sfârșitul fișierului) și vom contoriza câte afișări au fost efectuate până la momentul respectiv, în variabila c . Afișăm numărul imediat după citire pe aceeași linie ca și precedentul și urmat de un spațiu, iar în cazul în care au fost efectuate 5 afișări (dacă este 5) se trece la linie nouă și se resetează contorul c (c redevine 0).

Varianta C/C++

```
#include<fstream.h>
void main()
{
    long nr;
    int c=0;
    ifstream f("bac.txt");
    while(f>>nr)
        {    cout<<nr<<" ";
          c++;
          if(c==5)
              {    cout<<endl;
                c=0;}
        }
    f.close();
}
}
```

3. Varianta PASCAL

```
var f:text;
    nr:longint;
begin
    f:=integer;
    assign(f,'bac.txt');
    reset(f);
    c:=0;
    while not eof(f) do
        begin
            readln(f,nr);
            write(nr,' ');
            c:=c+1;
            if c=5 then
                begin
                    writeln;
                    c:=0;
                end;
        end;
    close(f);
end;
```

| end.

4. Pentru intensiv

Vom citi toate datele din fișierul de intrare (până la sfârșitul fișierului) și vom contoriza câte afișări au fost efectuate până la momentul respectiv, în variabila c. Afișăm numărul imediat după citire pe aceeași linie ca și precedentul și urmat de un spațiu, iar în cazul în care au fost efectuate 5 afișări (dacă este 5) se trece la linie nouă și se re setează contorul c (c redevine 0). Tot în momentul citirii vom determina și câte dintre numerele din fișier au suma cifrelor pară, utilizând funcția definită anterior pentru a calcula suma cifrelor unui număr natural.

Varianta C/C++

```
#include<fstream.h>
int sumac(long n)
{   int s=0;
    while(n)
        {s=s+n%10;
         n=n/10;}
}
return s;
void main()
{   long nr;
    int c=0,cp=0;
    ifstream f("bac.txt");
    while(f>>nr)
        {
            cout<<nr<<" ";
            c++;
            if(c==5)
                {
                    cout<<endl;
                    c=0;
                }
            if(sumac(nr)%2==0)
                cp++;
        }
    cout<<endl<<cp;
    f.close();
}
```

Varianta PASCAL

```
var f:text;
    nr:longint;
    c,cp:integer;
function sumac(n:longint):integer;
var s:integer;
begin
    s:=0;
    while n<>0 do
        begin
            s:=s+n mod 10;
            n:=n div 10;
        end;
    sumac:=s;
end;
begin
    assign(f,'bac.txt');
    reset(f);
    c:=0;cp:=0;
    while not eof(f) do
        begin
            readln(f,nr);
            write(nr,' ');
            c:=c+1;
            if c=5 then
                begin
                    writeln;
                    c:=0;
                end;
            if sumac(nr) mod 2=0
            then
                end;
                cp:=cp+1;
            writeln;
            write(cp);
            close(f);
        end;
end.
```

Pentru neintensiv

a) Verifică dacă numărul primit ca parametru are vreun divizor propriu, caz în care funcția returnează valoarea 0. Dacă numărul găsește niciun divizor atunci returnează 1.

b) Varianta C/C++

```
#include<iostream.h>
#include<math.h>
int prim(int n)
{   if (n<=1)
        return 0;
    for(int d=2;d<=sqrt(n);d++)
```

b) Varianta PASCAL

```
var nr:longint;
    i:integer;
function prim(n:longint):integer;
var d,p:integer;
begin
    p:=1;
```

```

        if(n%d==0)
            return 0;
    return 1;
}
void main()
{
for(int i=11;i<=97;i+=2)
if(prim(i) && prim(i%10*10+i/10))
    cout<<i<<" ";
}

```

```

if n<=1 then p:=0
else
    for d:=2 to n div 2 do
        if n mod d=0 then
            p:=0;
    prim:=p;
end;
begin
for i:=11 to 97 do
if (prim(i)=1)and(prim(i mod 10
*10+i div 10)=1) then
    write(i, ' ');
end.

```

Varianta <15>

SUBIECTUL I

1. d

2. a) 4

b) Varianta C/C++

```

#include<iostream.h>
#include<stdio.h>
void main()
{ unsigned i,n; int a;
scanf("%d",&a,&n); //cin>>a>>n;
for(i=1;i<=n;i++)
if(i%2==0) a=a-i*i;
    else a=a+i*i;
printf("%d",a); //cout<<a;}

```

c) Se observă că suma calculată la punctul (a) provine din $25+1^2-2^2+3^2-4^2+5^2-6^2$. Dacă facem excepție de valoarea variabilei **a** atunci suma rămasă este egală cu $(-1)^{n+1} * \frac{n(n+1)}{2}$ unde **n** este numărul de termeni ai sumei, în cazul nostru **n** este

egal cu valoarea 6. Pentru variabila **a=18**, ca să obținem la final valoarea 8, înseamnă că, din valoarea inițială a variabilei **a**, trebuie să scădem valoarea 10. Rezolvând acum ecuația se va obține pentru **n** valoarea 4.

d) Cerința este ambiguă. Respectând strict cerința atunci ar trebui să facem următoarea înlocuire:

```

dacă (i%2=0) atunci a=0;
altfel a=n*n;

```

Varianta PASCAL

```

program bac;
var i,n,a: integer;
begin
readln(a,n);
for i:=1 to n do
if (i mod 2 =0) then a:=a-i*i
else a:=a+i*i;
write(a);
end.

```

SUBIECTUL II

1. a pentru intensiv și d pentru neintensiv

2. d

3.9 bemeut

4. gradul minim este 2 iar nodurile care au gradul 2 sunt 5, 7, 8.

5. Pentru intensiv Se parcurge matricea

Pentru neintensiv Se parcurge

integral și se condiționează completarea
diagonalei principale cu valoarea 0, iar
pentru celelalte elemente completarea

matricei integral. După completarea
unei linii cu suma divizorilor săi
acesta își va schimba valoarea în

valorilor va ține seam de faptul că matricea este simetrică față de diagonala principală deci $a[i][j]=a[j][i]$. După completarea unei linii cu ultima cifră a

$n=n/10$.

numărului n , acesta își va schimba valoarea în $n=n/10$.

Varianta C/C++

Program intensiv

```
main()
{ unsigned m,a[7][7],n, i,j,k;
do{scanf("%d",&n);//cin>>n;
k=0;m=n;
while(m!=0){m=m/10;k++;}
}while(k!=5);
for(i=1;i<=6;i++)
{for(j=1;j<=6;j++)
if(i==j) a[i][j]=0;
else if(i<j)
a[i][j]=a[j][i]=n%10;
n=n/10;}
for(i=1;i<=6;i++)
{for(j=1;j<=6;j++)
printf("%d",a[i][j]);}cout<<a[i][j]<<" ";
printf("\n");}cout<<endl;}}
```

Varianta PASCAL

Program intensiv

```
type mat=array[1..7,1..7] of
integer;
var a:mat;
m,n,i,j,k:integer;
begin
repeat
readln(n); k=0;m=n;
while(m<>0) do
begin
m:=m div 10; k:=k+1;
end;
until (k=5);
for i:=1 to 6 do
begin
for j:=1 to 6 do
if i=j then a[i,j]:=0 else if i<j
then begin a[i,j]:=n mod 10;
a[j,i]:=n mod 10;end;
n:=n div 10;
end;
for i:=1 to 6 do
begin
for j:=1 to 6 do
write(a[i,j], ' ');
writeln();
end; end.
```

Varianta C/C++

Pentru neintensiv

```
main()
{ unsigned m,a[6][6],n, i,j,k;
do{scanf("%d",&n);//cin>>n;
k=0;m=n;
while(m!=0){m=m/10;k++;}
}while(k!=5);
for(i=1;i<=5;i++)
{for(j=1;j<=5;j++)
a[i][j]=n%10;
n=n/10;}
for(i=1;i<=5;i++)
{for(j=1;j<=5;j++)
printf("%d",a[i][j]);}cout<<a[i][j]<<" ";
printf("\n");}cout<<endl;}}
```

Varianta PASCAL

Pentru neintensiv

```
type mat=array[1..6,1..6] of
integer;
var a:mat;
m,n,i,j,k:integer;
begin
repeat
readln(n); k=0;m=n;
while(m<>0) do
begin
m:=m div 10; k:=k+1;
end;
until (k=5);
for i:=1 to 5 do
begin
for j:=1 to 5 do
a[i,j]:=n mod 10;
n:=n div 10; end;
for i:=1 to 5 do
begin
for j:=1 to 5 do
write(a[i,j], ' ');
```

```
writeln();
end; end.
```

SUBIECTUL III

1. b

2. 4 pentru intensiv și 85 pentru neintensiv

3. pentru intensiv: Pentru fiecare valoare din intervalul $[1, n]$, parcurs de la n la 1, se contorizează în variabila k numărul divizorilor și apoi se verifică cu valoarea \max (ce reprezintă numărul maxim de divizori). În cazul în care s-a găsit o valoare ce are un număr mai mare de divizori această valoare este testată cu

variabila \min (ce reprezintă cea mai mică valoare care nu depășește condiția pentru numărul maxim de divizori), și păstrată în \min dacă îndeplinește condiția impusă.

pentru neintensiv: Folosim o variabilă contor k pentru a număra de câte ori găsim cifra 5 în fiecare număr citit în variabila x . Pentru citire vom folosi o structură repetitivă cu test inițial care va include tot o astfel de structură pentru a parcurge numărul x , cifra cu cifră și a verifica existența cifrei 5. După fiecare verificare vom folosi operatorul \div pentru a elimina

cifra unităților și pentru a verifica următoarea cifră, a zecilor care, a devenit astfel cifra unităților. Înainte de a ne întoarce în prima structură repetitivă, citim următoarea valoare din șir.

Varianta C/C++**Pentru intensiv**

```
#include<iostream.h>
#include<stdio.h>
main()
{ unsigned n,min=32000,max=0,i,j,k;

do scanf("%d",&n);//cin>>n;
while(n>32000);
for(i=n;i>=1;i--)
{ k=0;
for(j=1;j<=i/2;j++)
if(i%j==0) k++;
if(max<=k) { max=k;
if(min>i) min=i;}
}
printf("%d",min);//cout<<min;
}
```

Pentru neintensiv:

```
#include<iostream.h>
#include<stdio.h>
main()
{unsigned long x; int k=0;
scanf("%d",&x);//cin>>x;
while(x!=0)
{ while(x!=0)
{if(x%10==5) k++; x=x/10;}
scanf("%d",&x);//cin>>x;}
printf("%d",k);//cout<<k;
}
```

4. a) **Pentru intensiv:**

Am folosit subprogramul prim care verifică dacă valoarea transmisă în

Varianta PASCAL**Pentru intensiv**

```
var n,min,max,i,j,k:integer;
begin
min:=32000;max:=0;
repeat readln(n); until(n<=32000);
for i:=n downto 1 do
begin
k:=0; for j:=1 to i div 2 do
if(i mod j =0) then inc(k);
if (max<=k) begin
max:=k; if(min>i) then min:=i;
end;end;
write(min);
end.
```

Pentru neintensiv:

```
var x:longint; k:integer;
begin
readln(x);k:=0;
while not x =0
begin
while not x =0
begin
if x mod 10=5 then inc(k);
x= x div 10;
end;
readln(x);end;
write(k);end.
```

Pentru neintensiv:

Citirea valorilor s-a realizat testând finalul de fișier. Ultima valoare impară

parametrul x, reprezintă un număr prim. Vom citi numerele din fișier, rand pe rand în variabila x (citirea se finalizează în momentul în care s-a ajuns la sfârșitul

fișierului). Variabilele a și b vor memora penultimul și respectiv ultimul număr prim citit din fișier, dacă acestea există. La citirea fiecărei valori se verifică dacă este număr prim și dacă noua valoare citită este diferită de ultima valoare memorată în variabila b. În caz afirmativ variabila a preia vechea valoare a variabilei b iar variabila b preia noua valoare a numărului prim citit în x.

b) Varianta C/C++

Program intensiv

```
#include<fstream.h>
ifstream f("BAC.IN");

int prim (long int x)
{long int i;
for(i=2;i<=sqrt(x);i++)
if(x%i==0) return 0;
return 1;}

main()
{int a=0,b=0,x;
while(f>>x)
if(prim(x)&& x!=b) {a=b;b=x;}
if(a*b>0) cout<<a<<" "<<b;
else cout<<"numere prime
insuficiente";
f.close();
}
```

Program neintensiv

```
#include<fstream.h>
ifstream f("BAC.IN");

main()
{ unsigned long k,x;int ok=0;
while(f>>x)
if(x%2!=0) {k=x;ok=1;}
if(ok) cout<<k;
else cout<<"nu exista numere impare";
f.close();
}
```

o păstrăm în variabila k. Folosim variabila logică ok pentru a verifica dacă există sau nu măcar o valoare impară în fișier, iar la finalul

programului, afișăm rezultatul în funcție de valoarea acestei variabile.

Varianta PASCAL

Program intensiv

```
var f:text;
a,b,x:longint;
function prim(x:longint):integer;
begin
var i:longint;
for i:=2 to i sqrt(x) div 2 do
if x mod i=0 then prim:=0;
prim:=1;
end;

begin
a:=0;b:=0;
assign(f,'BAC.IN');
reset(f);
while not eof(f) do
begin
read(f,x);
if (prim(x) and x<>b) then
begin
a:=b;b:=x;end;end;
if a*b>0 write(a,' ',b) else
write("numere prime insuficiente");
close(f);
end.
```

Program neintensiv

```
var f:text;
k,x:longint;ok:boolean;
begin
ok:=false;
assign(f,'BAC.IN');
reset(f);
while not eof(f) do
begin
read(f,x);
if (x mod 2 <>0) then
begin
k:=x;ok:=true;end;end;
if ok write(k) else write('nu
exista numere impare');close(f);
```

end.

Varianta <16>

SUBIECTUL I

1. a

2. a) un caracter

b) **Cerința este greșită.** Dacă se dorește afișarea caracterului # atunci răspunsul corect este 134.

c)

Varianta C/C++

```
void main()
{
    int n;
    scanf("%d", &n); //cin>>n;
    for(i=1; i<=n-1; i++)
        if(i%2==0) printf("#"); //cout<<'#';
    for(j=i+1; j<=n; j++)
        printf("*"); //cout<<'*';
}
```

d) citește n;

i=1;

cât timp (i<=n-1) execută
dacă(i%2=0) atunci scrie '#';

i=i+1;

j=i+1;

cât timp(j<= n) execută
scrie '*'; j=j+1;

Varianta PASCAL

```
var i, j, n: integer;
begin
    readln(n);
    for i:=1 to n-1 do if i mod 2=0
    then write('#');
    for j:=i+1 to n do write('*');
end.
```

SUBIECTUL II

1. b

2. a

3. pentru intensiv

$(x \rightarrow urm + x \rightarrow urm \rightarrow nr) / 2 \geq 4.75$

4. clasa a-XII-a A

CLASA A-XII-A A

5.

Varianta C/C++

Pentru intensiv

```
#include<iostream.h>
#include<stdio.h>
main()
{ unsigned a[20][20], n, i, j;
  do scanf("%d", &n); //cin>>n;
  while (n<=2 || n>=16);
```

pentru neintensiv : Se extrag două
elemente iar elemental din vârful stivei va
fi după extragere 7

clasa a-XII-a A

clasa aa

Varianta PASCAL

Pentru intensiv

```
type mat=array[1..20,1..20] of
integer;
var a:mat; n,i,j:integer;
begin
repeat
readln(n);
```



```

for(i=1;i<=n;i++)
for(j=1;j<=n;j++)
    if (i==j || i+j==n+1)
a[i][j]=0;
    else if (i<j && i+j<n+1)
a[i][j]=1;
    else if (i>j && i+j>n+1)
a[i][j]=2;
    else a[i][j]=3;
for(i=1;i<=n;i++)
{for(j=1;j<=n;j++)
printf("%d
",a[i][j]);}cout<<a[i][j]<<" ";
printf("\n");}cout<<endl;}
}

```

Pentru neintensiv

```

#include<iostream.h>
#include<stdio.h>

main()
{ unsigned a[16][16],n, i,j;
do scanf("%d",&n);}cin>>n;
while(n<=2||n>=16);
for(i=1;i<=n;i++)
for(j=1;j<=n;j++)
    if (i==j || i+j==n+1)
a[i][j]=4;
    else a[i][j]=3;
for(i=1;i<=n;i++)
{for(j=1;j<=n;j++)
printf("%d
",a[i][j]);}cout<<a[i][j]<<" ";
printf("\n");}cout<<endl;}
}

```

SUBIECTUL III

1. d

2. (7,7,7,7,3) (7,7,7,7,5) (7,7,7,7,7)

3.

Varianta C/C++

Pentru intensiv

```

int multiplu(int a[100],int n, int
k)
{ int i,p=0;
for(i=1;i<=n;i++)
if(a[i]%k==0 && a[i]%10==k) p++;
return p;
}

```

Pentru neintensiv:

```

#include<iostream.h>
#include<stdio.h>

```

```

until(n>2 or n<20);
for i:=1 to n do
for j:=1 to n do
if (i=j or i+j=n+1) then a[i,j]:=0
else if(i<j and i+j<n+1) then
a[i,j]:=1
else if(i>j and i+j>n+1) then
a[i,j]:=2
else a[i,j]:=3;
for i:=1 to n do
begin
for j:=1 to n do
write(a[i,j], ' ');
writeln();
end; end.

```

Pentru neintensiv

```

type mat=array[1..16,1..16] of
integer;

var a:mat; n,i,j:integer;
begin
repeat
readln(n);
until(n>2 or n<20);
for i:=1 to n do
for j:=1 to n do
if (i=j or i+j=n+1) then a[i,j]:=4
else a[i,j]:=3;
for i:=1 to n do
begin
for j:=1 to n do
write(a[i,j], ' ');
writeln();
end; end.

```

Varianta PASCAL

Pentru intensiv

```

type vector=array[1..100] of
integer;
function multiplu(a:vector,
n:integer, k:integer):integer;
begin
var i,p:integer;
p:=0;
for i:=1 to n do
if (a[i] mod k=0 and a[i] mod 10=k)
inc(p);
multiplu:=p;
end;

```

Pentru neintensiv:

```

type vector=array[1..10] of integer;
var v:vector; i,k:integer;

```

```
main()
{unsigned v[10],i,k=0;
for(i=1;i<=10;i++)
{scanf("%d",&n);//cin>>v[i];
if(v[i]%13==0) k++;}
printf("%d ",k);//cout<<k<<" ";
for(i=1;i<=10;i++)
if(v[i]%13==0) printf("%d
",i);//cout<<i<<" ";
```

```
begin
k:=0;
for i:=1 to 10 do
begin
readln(v[i]); if v[i] mod 13=0 then
inc(k); end;
write(k,' ');
for i:=1 to 10 do
if v[i] mod 13=0 then write(i,' ');
end.
```

4. a) pentru intensiv

O idee de rezolvare a acestei probleme constă în citirea tuturor numerelor din fișier într-o singură variabilă x , și contorizarea cifrelor fiecărui număr citit într-un vector de zece elemente, în care pozițiile elementelor reprezintă cifrele de la 0 la 9. La finalul citirii din fișier se vor afișa din vectorul v , numai valorile în care s-a

contorizat pozițiile. Pentru acest lucru nu s-a folosit decât o singură variabilă x și variabilă j care permite numărul de apariții a cifrei respective și afișarea ei de câte ori a fost găsită. Și din punct de vedere al gestionării memoriei acest algoritm este cel mai eficient, pentru că nu memorează cele 10000 de numere cu maxim nouă cifre, într-un vector, folosind pentru acest lucru o singură variabilă, adică x .

b) Varianta C++

Pentru intensiv

```
#include<iostream.h>
#include<stdio.h>

ifstream f("numere.txt");
int v[10];

main()
{ long x;int i,j;
while (f>>x)
while(x!=0){v[x%10]++;x=x/10;}
for (i=9;i>=0;i--)
for(j=1;j<=v[i];j++) cout<<i;
f.close();
}
```

Pentru neintensiv:

```
#include<iostream.h>
#include<stdio.h>

ifstream f("numere.txt");
int cifrak(unsigned long n, unsigned
k)
{int p=0;
while(n!=0){if (n%10==k)
p++;n=n/10;}

return p;}

main()
{ unsigned long n;
while (f>>n)
```

Varianta PASCAL

Pentru intensiv

```
type vector=array[1..10] of byte;
var f:text; v:vector;
x:longint; i,j:integer;
begin
assign(f,'numere.txt');
reset(f);
while not eof(f) do
begin
read(f,x);
while(x<>0)
begin
inc(v[x mod 10]); x:=x div 10; end;
for i:=9 downto 0 do
for j:=1 to v[i] do
write(i);

close(f);
end.
```

Pentru neintensiv:

```
var f:text; n:longint;
function cifrak(longint n, k
integer):integer;
var p:integer;
begin
p:=0;
while n<>0 do begin
if n mod 10=k inc(p); n:=n div
10;end;

return p;end;

assign(f,'numere.txt');reset(f);
read(f,n);
```

```
if(cifrak(n,0)==3) cout<<n<<" ";
f.close();
}
```

```
while not eof(f) do begin
if cifrak(n,0)=3 write(n,' ');
readln(f,n);end;
close(f);end.
```

SUBIECTUL I

1. b

2. a) ABABABAB

b) 8

c)

Varianta C/C++

```
void main()
{ unsigned x,y;
scanf("%d%d",&x,&y);//cin>>x>>y;
if(x<y) {x=x-y;y=x+y;x=y-x;}
while(x>=y)
{ printf("A");//cout<<'A';
x=x-y;
printf("B");//cout<<'B';
}}
```

d)

```
┌repetă
│  scrie 'A';
│  x=x-y;
│  scrie 'B';
└pânăcând x<y;
```

Varianta <17>

Varianta PASCAL

```
var x,y: integer;
begin
readln(x,y);
if x<y then
begin x:=x-y;y:=x+y;x:=y-x;end;
while(x>=y) do
begin
write("A");x:=x-y;write("B");
end;end.
```

SUBIECTUL II

1. a “ “

2. b

3. pentru intensiv: maxim este 4, minim este 1, pentru neintensiv: 3 și 1

4. pentru intensiv $11*****$ pentru neintensiv: $11b*t$

5. Se parcurge matricea integral, valoarea elementelor de pe o linie a matricei va fi egala cu variabila k , ce primește pe fiecare linie ca valoare indicele liniei respective, iar după atribuirea $a[i][j]=k$ variabila k se incrementează cu o unitate.

Varianta C/C++

```
main()
{ unsigned a[20][20],n, i,j,k;
do scanf("%d",&n);//cin>>n;
while(n<=2||n>=20);
for(i=1;i<=n;i++)
{k=i;
for(j=1;j<=n;j++)
{a[i][j]=k;k++;}}
for(i=1;i<=n;i++)
{for(j=1;j<=n;j++)
printf("%d",a[i][j]);
//cout<<a[i][j]<<" ";
printf("\n");//cout<<endl;}
```

Varianta PASCAL

```
type mat=array[1..20,2..20] of
integer;
var a:mat; n,i,j,k:integer;
begin
repeat
readln(n);
until(n>2 or n<20);
for i:=1 to n do
begin
k:=i;
for j:=1 to n do
begin
a[i,j]:=k;inc(k); end;
```

```

}
end;
for i:=1 to n do
begin
for j:=1 to n do
write(a[i,j], ' ');
writeln();
end; end.

```

SUBIECTUL III

1. a

2. 12347, 12346, 12345

3. Se vor compara cele două valori și se va salva în **min** cea mai mică și în **max** cea mai mare. Se parcurge vectorul de la a doua poziție până la penultima (observați că variabila contor **k** a plecat de la valoarea 2, adică primul și ultimul element au fost deja numărate) și se verifică condițiile impuse în enunțul problemei contorizând în variabila **k** numărul de elemente ce verifică condițiile.

Varianta C/C++

Pentru intensiv

```

int interval(int v[100], int n)
{ int i, k=2, min, max;
if (v[1]>v[n]) {min=v[n]; max=v[1];}
else {min=v[1]; max=v[n];}
for (i=2; i<=n-1; i++)
if (v[i]>=min && v[i]<=max) k++;
return k; }

```

Pentru neintensiv:

Varianta PASCAL

Pentru intensiv

```

type vector=array[1..100] of
integer;
function interval(a:vector,
n:integer):integer;
begin
i, k, min, max:integer;
k:=2;
if v[1]>v[n] then
begin min:=v[n]; max:=v[1]; end
else begin min:=v[1]; max:=v[n]; end;
for i:=2 to n-1 do
if (v[i]>=min and v[i]<=max) then
inc(k);
interval:=k;
end;

```

4 a) O idee de rezolvare a acestei probleme constă în citirea separată a primului număr din șir în variabila **x**, apoi toate celelalte numere vor fi citite numai în variabila **y**, verificând la fiecare citire dacă numărul citit este mai mic decât cel reținut în variabila **x**. În acest fel, doar numărăm câte numere din șir sunt mai mici decât primul, și evităm citirea numerelor într-un vector (economie de memorie) și ordonarea vectorului (minimizând timpul de execuție).

b) Varianta C/C++

Pentru intensiv

```

FILE *f=fopen("numere.txt");//
ifstream f("numere.txt");
main()
{ unsigned x, y, i, n, k=0;
fsacnf(f, "%d%d", &n, &x); //f>>n>>x;
for(i=1; i<=n-1; i++)
{ fsacnf(f, "%d", &y); //f>>y; if(y<x)
k++;}
printf("%d", k); // cout<<k;
fclose(); //f.close();
}

```

Varianta PASCAL

Pentru intensiv

```

var f:text;
x, i, y, n, k:integer;
begin
assign(f, 'numere.txt');
reset(f); read(f, n, x);
for i:=1 to n-1 do
begin
read(f, y); if y<x then inc(k);
end;
write(k);
close(f);

```

Pentru neintensiv

```
ifstream f("numere.txt");

main()

while (v[1] <= v[n]) i=i+1;
n=i-1;
if (v[1] <= v[n])
printf("%d", interval(v,n));
//cout<<interval(v,n);
else printf(1); //cout<<1;
fclose(); //f.close();
}
```

end.

Pentru neintensiv

```
type vector=array[1..100] of
integer;
var f:text,v:vector;

begin integer;
assign(f,'numere.txt');
reset(f); i:=2; read(f,v[1]);
while not eof(f) begin
read(f,v[i]); inc(i); end;
n:=i-1;
if v[1] <= v[n] then
write(interval(v,n)) else write(1);
close(f);
end.
```

SUBIECTUL I

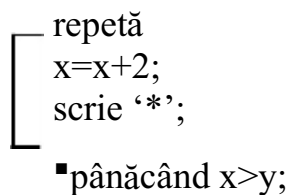
- 1. a
- 2. a) ****
- b) 1 și 0
- c)

Varianta C/C++

```
void main()
{ unsigned x,y;
scanf("%d",&x,&y); //cin>>x>>y;

if (x>y) {x=x-y;y=x+y;x=y-x;}
if (x%2==0) x=x+1;
while (x<=y)
{ x=x+2;
printf("**"); //cout<<'*';
}
}
```

d)



SUBIECTUL II

- 1. pentru intensiv: b, pentru neintensiv: c
- 2. a
- 3. pentru intensiv: înălțimea minimă este 3 iar noduri terminale sunt 4
pentru neintensiv: 4 și 4
- 4. if (strchr("aeiou",a[i])!=NULL) cout<<'*'; else cout<<a[i];
- 5. Se parcurge matricea integral și se completează impunând condiție pentru indicele de linie.

Varianta C/C++

```
main()
{ unsigned a[20][20],n, i,j;
do scanf("%d",&n); //cin>>n;
```

Varianta PASCAL

```
type mat=array[1..20,2..20] of
integer;
var a:mat; n,i,j:integer;
```

Varianta <18>

```
while (n<=2 | n>=20);
for (i=1; i<=n; i++)
for (j=1; j<=n; j++)
if (i%2!=0) a[i][j]=i; else
a[i][j]=j;
for (i=1; i<=n; i++)
{for (j=1; j<=n; j++)
printf("%d
", a[i][j]); //cout<<a[i][j]<<" ";
printf("\n"); //cout<<endl;}
}
```

```
begin
repeat
readln(n);
until (n>2 or n<20);
for i:=1 to n do
for j:=1 to n do
if (i mod 2<>0) then a[i,j]:=i else
a[i,j]:=j;
for i:=1 to n do begin
for j:=1 to n do
write(a[i,j], ' ');
writeln(); end; end.
```

SUBIECTUL III

1. b

2. 11101, 11110, 11111

3. **pentru intensiv:** Se calculează suma elementelor din tablou, apoi se împarte la numărul de elemente pentru a afla media aritmetică, și apoi se parcurge tabloul integral pentru a compara fiecare element cu media aritmetică, iar în cazul în care găsim elemente mai mari sau egale cu media aritmetică incrementăm valoarea contorului k.

Varianta C/C++

Pentru intensiv

```
int count(float v[100], int n)
{ float s=0, ma; int i, k=0;
for (i=1; i<=n; i++) s=s+v[i];
if (n!=0) ma=s/n;
for (i=1; i<=n; i++) if (v[i]>=ma)
k++;
return k; }
```

Pentru neintensiv

```
int count(float v[100], int n)
{ float ma=(v[1]+v[n])/2; int
i, k=0;
for (i=2; i<=n-1; i++)
if (v[i]>=ma) k++;
return k; }
```

Varianta PASCAL

Pentru intensiv

```
type vector=array[1..100] of integer;
var v:vector;
function count(v:vector,
n:integer):integer;
begin
var i, k:integer; s, ma:real;
k:=0; s:=0;
for i:=1 to n do s:=s+v[i];
if n<>0 then ma:=s/n;
for i:=1 to n do if (v[i]>=ma) then
inc(k);
count:=k;
end;
```

Pentru neintensiv

```
type vector=array[1..100] of real;
var v:integer;
function count(v:vector,
n:integer):integer;
begin
var i, k:integer; ma:real;
k:=0; ma=(v[1]+v[n])/2;
for i:=2 to n-1 do if (v[i]>=ma) then
inc(k);
count:=k;
end;
```

4. a) Ideea de rezolvare este evitarea memorării numerelor din șir într-un vector și ordonarea șirului descrescător. Numărăm câte elemente din șir sunt mai mari decât numărul k, și realizăm economie de memorie și evitând ordonarea vectorului

minimizăm timpul de execuție.

b) Varianta C/C++

```
ifstream f("numere.txt");
main()
{ unsigned x,k,p=0,ok=0;
cin>>k;
while (f>>x)
{if(x>=k) p++; if(x==k)ok=1;}
if(ok)cout<<p;
else cout<<"nu exista";
f.close();
}
```

```
ifstream f("numere.txt");
main()
{ unsigned k=0,i=1,n; float v[90];
while (f>>v[i]) i++;
n=i-1;
if(v[1]==0) cout<<count(v,n);
else{ for(i=1;i<=n-1;i++)
if(v[i]>=v[n]/2) k++; cout<<k;}
f.close();
}
```

Varianta PASCAL

```
var f:text;
x,p,k:integer;
ok:boolean;
begin
assign(f,'numere.txt');
reset(f); readln(k);read(f,x);
p:=0;ok:=false;
while not eof(f) do
begin
if x>=k then inc(p);
if x=k then ok:=true; read(f,x);
end;
if ok=true then write(p) else
write("nu exista");
close(f);end.
```

```
type vector=array[1..90] of real;
var f:text; v:vector;
n,i,k:integer;
begin
assign(f,'numere.txt');k:=0;i:=2;
reset(f); read(f,v[1]);
while not eof(f) do
begin
read(f,v[i]);inc(i);
end;
if v[1]=0 then write(count(v,n))
else begin
for i:=1 to n-1 do
if v[i]>=v[n/2] inc(k);
write(k);end;
close(f);end.
```

Varianta <19>

SUBIECTUL I

1. a

2. a) 234

b) Ultimele două cifre ale variabilelor **a** și **b** trebuie să reprezinte capetele unui interval de numere naturale în care să se găsească exact două numere cu cifrele identice. Exemplu: a=521, b=1234. Atenție: ultimele două cifre ale variabilei **a** trebuie să reprezinte un număr mai mic decât numărul format din ultimele două cifre ale variabilei **b**.

c)

Varianta C/C++

```
void main()
{ unsigned a,b,i;
scanf("%d%d",&a,&b); //cin>>a>>b;
a=a/10%10*10+a%10;
b=b/10%10*10+b%10;
for(i=a;i<=b;i++) if(i/10==i%10)
printf("%d", i%10); //cout<<i%10;
}
```

Varianta PASCAL

```
var a,b,i:integer;
begin
readln(a,b);
a:=a div 10 mod 10*10+a mod 10;
b:=b div 10 mod 10*10+b mod 10;
for i:=a to b do
if i div 10=i mod 10 then write(i
mod 10);
end.
```

d. `i=a;`
 [`cât timp (i<=b) execută`
 `dacă[i/10]=i%10 atunci`
 `scrie i%10;`
] `i=i+1;`

SUBIECTUL II

1. a

2. c

3. pentru intensiv bac2008

pentru neintensiv2008

bac

bac2008

4. pentru intensiv `a.y==b.y` pentru neintensiv primul element: 2, ultimul element : 7

5. pentru intensiv : Se parcurge matricea integral. Se verificădac aîndicele de linie

este impar, și dacăda, atunci, elementele de pe respectiva linie primesc ca valoare suma dintre indicele de linie și indicele de coloană. In caz contrar toate elementele primesc valoarea celui de pe linia precedentădin stânga pozi ției celei pe care o completăm la momentul respectiv, valoarea acestuia fiind cea mai micădintre cele trei, vecine, în afarăde cel de pe prima coloan ă, care va fi permanent egal cu indicele de linie, acesta din urmăfiind cea mai mic ăvaloare dintre cele două ăvecine.

Varianta C/C++

Pentru intensiv:

```
unsigned a[20][20];
main()
{ unsigned n, i, j, min;
do scanf("%d", &n); //cin>>n;
while (n<=2 || n>=20);
for (i=1; i<=n; i++)
for (j=1; j<=n; j++)
if (i%2!=0) a[i][j]=i+j;
else
if (j==1) a[i][j]=i; else
a[i][j]=a[i-1][j-1];
for (i=1; i<=n; i++)
{for (j=1; j<=n; j++)
printf("%d
", a[i][j]); //cout<<a[i][j]<<" ";
printf("\n"); //cout<<endl;}}
```

Pentru neintensiv:

```
unsigned a[20][20];
main()
{ unsigned n, i, j, min;
do scanf("%d", &n); //cin>>n;
while (n<=2 || n>=20);
for (i=1; i<=n; i++)
for (j=1; j<=n; j++)
if (j%2!=0) a[i][j]=i+j;
else a[i][j]=i;
for (i=1; i<=n; i++)
{for (j=1; j<=n; j++)
printf("%d", a[i][j]); //
cout<<a[i][j]<<" ";
printf("\n"); //cout<<endl;}}
```

Varianta PASCAL

Pentru intensiv:

```
type mat=array[1..20,2..20] of
integer;
var a:mat; n,i,j,min:integer;
begin
repeat
readln(n);
until (n>2 or n<20);
for i:=1 to n do
for j:=1 to n do
if i mod 2<>0 then a[i,j]:=i+j
else if j=1 then a[i,j]:=i
else a[i,j]:=a[i-1,j-1];
for i:=1 to n do begin
for j:=1 to n do
write(a[i,j], ' ');
writeln();end;end.
```

Pentru neintensiv:

```
type mat=array[1..20,2..20] of
integer;
var a:mat; n,i,j,min:integer;
begin
repeat
readln(n);
until (n>2 or n<20);
for i:=1 to n do
for j:=1 to n do
if j mod 2<>0 then a[i,j]:=i+j
else a[i,j]:=i;
for i:=1 to n do begin
for j:=1 to n do
write(a[i,j], ' ');
```



```
| writeln();end;end.
```

SUBIECTUL III

1. a

2. 10349, 10356, 10357

3.

Varianta C/C++

Pentru intensiv

```
void aranjare(float a[100], int n)
{int i,j; float aux;
for(i=1;i<=n-1;i++)
for(j=i+1;j<=n;j++)
if(a[i]>a[j])
{aux=a[i];a[i]=a[j];a[j]=aux;}
}
```

Pentru neintensiv:

Varianta PASCAL

Pentru intensiv

```
type vect=array[1..100] of real;
procedure aranjare (a:vect, n:integer);
begin
var i,j:integer; aux:real;
for i:=1 to n-1 do
for j:=i+1 to n do
if a[i]>a[j] then begin
aux:=a[i];a[i]:=a[j];a[j]:=aux; end;
end;
```

4 a) Pentru intensiv: Ideea de rezolvare constă în citirea pe rând din cele două fișiere a valorilor, cele din nr1 se vor citi în x, cele din nr2 în y. Prin evitarea citirii lor într-un vector, se utilizează eficient memoria. Din punct de vedere a timpului de execuție acest algoritm este mult mai eficient pentru că ocolește ordonarea valorilor.

Algoritmul este asemănător celui de interclasare a doi vectori ordonați. În variabila „a” am memorat permanent ultima valoare afișată. Înainte de a afișa o valoare nouă care îndeplinește condiția (de a fi mai mică decât ultima citită) se verifică dacă această valoare nu a mai fost afișată anterior, evitând astfel afișarea dublurilor. La finalul programului în cazul în care nu avem un același număr de numere în cele două fișiere, și știind că valorile din fiecare fișier sunt distincte și ordonate crescător, am făcut verificarea finalului de fișier și în cazul în care au mai rămas valori necitite le-am afișat exact în ordinea citirii lor.

b) Varianta C++

Pentru intensiv

```
ifstream f("nr1.txt"),g("nr2.txt");
main()
{ unsigned x,y,a;
long n1,n2,k=0,p=0;
f>>n1>>x;g>>n2>>y;
while(k<=n1&&p<=n2)
if(x<y)
{if(x!=a)cout<<x<<"
";f>>x;a=x;k++;}
else
{if(y!=a)cout<<y<<" ";g>>y;a=y;p++;}
while(!f.eof())
{cout<<x<<" ";f>>x;}
while(!g.eof())
{cout<<y<<" ";g>>y;}

else;};
```

Varianta PASCAL

Pentru intensiv

```
var f,g:text;
x,y,a:integer;n1,n2,k,p:longint;
begin
assign(f,'nr1.txt');assign(g,'nr2.txt');
reset(f);reset(g);
read(f,n1,x);read(g,n2,y); p:=0;k:=0;
while(k<=n1 and p<=n2)
if x<y then
begin
if x<a then write(x)
read(f,x); a:=x;inc(k);
end
else
begin if y<a write(y); read(g,y);
inc(p);

end;
while not eof(f) do
begin write(x);readln(f,x); end;
while not eof(g) do
```

Pentru neintensiv

```
ifstream f("nr1.txt");
ofstream g("nr2.txt");

main()
{
    int n,m;
    cin>>n>>m;
    for(i=1;i<=n;i++)
    for(j=1;j<=m;j++)
    {f>>a[i][j];a[i][j]=-a[i][j];}
    for(i=1;i<=n;i++)
    aranjare(a[i],m);
    for(i=1;i<=n;i++)
    for(j=1;j<=m;j++)
    a[i][j]=-a[i][j];
    for(i=1;i<=n;i++)
    {for(j=1;j<=m;j++) g<<a[i][j]<<" ";
    g<<endl;}
    f.close();g.close();}
}
```

```
begin write(y);readln(g,y); end;
close(f);close(g);end.
```

Pentru neintensiv

```
type mat=array[1..100,1..100] of real;
var f,g:text;

begin
    readln(n,m);
    assign(f,'nr1.txt');assign(g,'nr2.txt');
    reset(f);rewrite(g);
    read(f,n,m);
    for i:=1 to n do
    for j:=1 to m do
    begin
        read(f,a[i,j];a[i,j]:=-a[i,j];
    end;
    for i:=1 to n do aranjare(a[i],m);
    for i:=1 to n do
    for j:=1 to m do
    a[i,j]:=-a[i,j];
    for i:=1 to n do
    begin
        for j:=1 to m do
        write(g,a[i,j],' ');
        writeln(f);
    end;
    close(f);close(g);
end.
```

Varianta <20>

ȘUBIECTUL I

2. a) 9831

b) 1000

c) Varianta C/C++

```
void main()
{ unsigned n,a,m,b;
scanf("%d",&n);//cin>>n;
a=n%10;m=a;
while(n>9)
{ n=n/10; b=n%10;
if(a>b){m=m*10+b;a=b;}
}
printf("%d",m);//cout<<m;}
```

Varianta PASCAL

```
var n,a,m,b:integer;
begin
readln(n);
a:=n mod 10;
m:=a;
while(n>9) do
begin
n:=n div 10;b:=n mod 10;
if a>b then begin m:=m*10+b;a:=b;
end;
write(m);
end.
```

d) citește n

a=n%10; m=a;

dacă(n>9) atunci

repetă

n=[n/10];

b=n%10;

dacă a>b atunci

 m=m*10+b;

 a=b;

scrie m;

SUBIECTUL II

1. pentru intensiv:a , pentru neintensiv:d

2. pentru intensiv:d , pentru neintensiv:a

3. pentru intensiv: 5 , pentru neintensiv:9 cu 7
3

4. pentru intensiv: (strchr("0123456789",a[i])==0) , pentru neintensiv:
(i=strlen(a)-1;i>=0;i--)

5.

Varianta C/C++

Pentru intensiv

```
main()
{ unsigned a[20][20],n, i, j;
do scanf("%d",&n); //cin>>n;
while(n<=2 || n>=20);
for(i=1;i<=n;i++)
for(j=1;j<=n;j++)
if(i==1 || i==n || j==n || j==1)
a[i][j]=i+j;
else a[i][j]=a[i-1][j-1]+a[i-1][j]+a[i-1][j+1];
for(i=1;i<=n;i++)
{for(j=1;j<=n;j++)
printf("%d",a[i][j]);
printf("\n");
}
```

Pentru neintensiv:

```
main()
{ unsigned a[10][10],n, i, j;
do scanf("%d",&n); //cin>>n;
while(n<=2 || n>=10);
for(i=1;i<=n;i++)
for(j=1;j<=n;j++)
if(i==1 || j==1) a[i][j]=i+j;
else a[i][j]=a[i][j-1]+a[i-1][j];
for(i=1;i<=n;i++)
{for(j=1;j<=n;j++)
printf("%d ",a[i][j]);
//cout<<a[i][j]<<" ";
printf("\n"); //cout<<endl;}}
```

Varianta PASCAL

Pentru intensiv

```
type mat=array[1..20,1..20] of
integer;
var a:mat; n,i,j:integer;
begin
repeat readln(n); until(n>2 or
n<20);
for i:=1 to n do
for j:=1 to n do
if i=1 or i=n or j=1 or j=n then
a[i,j]:=i+j
else a[i,j]:=a[i-1,j-1]+a[i-1,j]+a[i-1,j+1];
for i:=1 to n do begin
write(a[i,j]);
writeln();end;end.
```

Pentru neintensiv:

```
type mat=array[1..10,1..10] of
integer;
var a:mat; n,i,j:integer;
begin
repeat readln(n); until(n>2 or
n<10);
for i:=1 to n do
for j:=1 to n do
if i=1 or j=1 then a[i,j]:=i+j
else a[i,j]:=a[i,j-1]+a[i-1,j];
for i:=1 to n do begin
for j:=1 to n do
write(a[i,j], ' ');
writeln();end;end.
```

SUBIECTUL III

1. c

2. 35789,35679,35678

3. Ideea de rezolvare constă în numărarea valorilor de 0 din vector iar apoi mutarea lor pe ultima poziție, utilizând la mutare o poziție suplimentară la care apoi renunțăm și permutăm la stânga cu o poziție elementele vectorului. ă

Varianta C/C++

Varianta PASCAL

```
void nule(int a[100],int n)
{ int i,j,k,nr=0;
for(i=1;i<=n;i++)
if(a[i]==0)nr++;
k=1;i=1;
while(k<=nr)
if(a[i]==0)
{n=n+1;
a[n]=a[i];
for(j=i;j<=n-1;j++) a[j]=a[j+1];
n--;k++;}
else i++;}
```

```
type vect=array[1..100] of integer;
procedure nule(a:vect,i:integer);
begin
var i,j,k,nr:integer;
for i:=1 to n do
if a[i]=0 then inc(nr);
k:=1;i:=1;
while k<=nr do
if a[i]=0 then begin
n:=n+1;a[n]:=a[i];
for j:=i to n-1 do a[j]:=a[j+1];
dec(n);inc(k);
end
else inc(i);end;
```

4 a) Pentru a rezolva în mod eficient problema a fost evitată folosirea tablourilor pentru memorarea valorilor din cele două fișiere, iar citirea și verificarea valorilor egale din cele două fișiere s-a făcut în același timp pentru amândouă, evitând astfel

metoda conform căreia: se citește prima valoare din fișier și se verifică cu toate din al doilea, șamd... Au fost folosite două variabile pentru citire x și y . Parcurgerea s-a realizat până la finalizarea valorilor dintr-un fișier impunând condiția $n1 > 0$ && $n2 > 0$, și decrementând cele două valori la fiecare citire ($n1$ se decrementează la citirea $f >> x$ iar $n2$ la citirea $g >> y$).

Varianta C/C++

Pentru intensiv

```
#include<fstream.h>
ifstream
f("nr1.txt"),g("nr2.txt");
main()
{ unsigned x,y,n1,n2;
f>>n1>>x;g>>n2>>y;n1--;n2--;
while(n1>0 && n2>0)
if(x==y) {cout<<x<<" ";f>>x;n1--;
g>>y;n2--;}
else if (x>y){ g>>y;n2--;}
else {f>>x;n1--;}
f.close();
g.close();
}
```

Pentru neintensiv

```
ifstream f("nr1.txt");
ofstream g("nr2.txt");
main()
{ int a[100][100], n,m,i,j,aux;
f>>n>>m
for(i=1;i<=n;i++)
for(j=1;j<=m;j++)
f>>a[i][j];

for(i=1;i<=n;i++)
for(j=1;j<=m/2;j++)
```

Varianta PASCAL

Pentru intensiv

```
var f,g:text;
x,y,n1,n2:integer;
begin
assign(f,'nr1.txt');assign(g,'nr2.txt');
reset(f);reset(g);
read(f,n1,x);read(g,n2,y);
while (n1>0 and n2>0) do
if x=y then
begin
write(x,' '); read(f,x);dec(n1);
read(g,y);dec(n2);
end
else if x>y then
begin
read(g,y);dec(n2);
end
else begin read(f,x);dec(n1);
end;
close(f);close(g);end.
```

Pentru neintensiv

```
type mat=array[1..100,1..100] of integer;
var f,g:text;
a:mat;i,n,j,m,aux:integer;
begin
assign(f,'nr1.txt');assign(g,'nr2.txt');
reset(f);rewrite(g);
read(f,n,m);
for i:=1 to n do
for j:=1 to m do
read(f,a[i,j]);
for i:=1 to n do nule(a[i],m);
for i:=1 to n do
```

<pre>{aux=a[i][j];a[i][j]=a[i][m+1-j]; a[i][m+1-j]=aux;} for(i=1;i<=n;i++) {for(j=1;j<=m;j++) g<<a[i][j]<<" "; g<<endl;} f.close();g.close();}</pre>	<pre>for j:=1 to m div 2 do begin aux:=a[i,j];a[i,j]:=a[i,m+1-j]; a[i][m+1-j]:=aux; end; for i:=1 to n do begin for j:=1 to m do write(g,a[i,j], ' '); writeln(f); end; close(f);close(g); end.</pre>
--	---

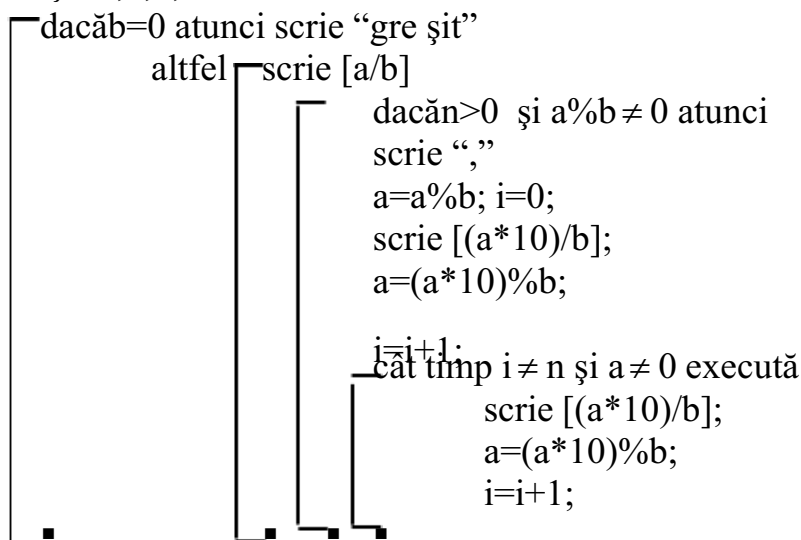
Varianta <21>

SUBIECTUL I

1. c

2. a) 2,8333

b) citește a,b,n;



c)

Varianta C/C++

```
#include<iostream.h>
#include<stdio.h>

main()
{signed a,b,n,i;
scanf("%d%d%d",&a,&b,&n);
//cin>>a>>b>>n;
if(b==0)printf("gresit");//cout<<"gresit";
else {printf("%d",a/b);//cout<<a/b;
if(n>0 && a%b!=0)
{printf(",");//cout<<',';
a=a%b;i=0;
do
{printf("%d", (a*10)/b);//cout<<(a*10)/b;
a=(a*10)%b;i++;}
while(i!=n&&a!=0);
}
}}
```

d) 15,4,4

Varianta PASCAL

```
program bac;
var n,a,i,b:integer;

begin
readln(a,b,n);
if b=0 then write("gresit")
else
begin
write(a div b);
if (n>0 and a mod b <>0) then
begin
write(",");a:=a mod b;i:=0;
repeat
write((a*10) div b); a:=(a*10)
mod b; i:=i+1;
until (i=n or a=0);
end;
end;end.
```

SUBIECTUL II

1. pentru intensiv: d pentru neintensiv: c

2. c

3. ev.data_nasterii.an=1990;

4. 4

5. Se parcurge șirul de caractere verificând dacă în două poziții consecutive se găsește același caracter cu condiția suplimentară ca acesta să fie diferit de caracterul '*'.

Varianta C/C++

```
main()
{ unsigned i=0; char a[30];
scanf("%s",&a);//cin.get(a,30);
for(i=0;i<strlen(a)-1;i++)
if(a[i]==a[i+1]&&a[i]!='*')
printf("%s%s",a[i],a[i]);//cout<<a[i]<<a[i]<<endl;
}
```

Varianta PASCAL

```
var i,n:integer;
a:string;
begin
readln(a);
n:=length(a);
for i:=0 to n-1 do
if (a[i]=a[i+1] and
a[i]<>'*') then
writeln(a[i],a[i]);
end.
```

SUBIECTUL III

1. c

2. 3

3.

Varianta C/C++

```
int i_prim(unsigned n)
{ int ok,gasit=0,i,k=n;
while(!gasit)
{ok=1;
for(i=2;i<=sqrt(k);i++) if(k%i==0)
ok=0;
if(ok) gasit=1; else k++;}
p2=k; k=n;gasit=0;
while(!gasit)
{ok=1;
for(i=2;i<=sqrt(k);i++) if(k%i==0)
ok=0;
if(ok) gasit=1; else k--;}

p1=k;
return p2-p1;
}
```

Varianta PASCAL

```
function i_prim(n:integer);
var i,k; ok,gasit:boolean;
begin
gasit:=false;k:=n;
while not gasit do
begin
ok:=true;
for i:=2 to sqrt(k) do if k mod i =0
then ok:=false;
if ok then gasit:=true else inc(k);
end;
p2:=k; k:=n;gasit:=false;
while not gasit do
begin
ok:=true;
for i:=2 to sqrt(k) do if k mod i =0
then ok:=false;
if ok then gasit:=true else dec(k);
end;
p1:=k; i_prim:=p2-p1;
end;
```

4 a)

Varianta C/C++

```
#include<fstream.h>
ifstream f("bac.txt");
int n,k,i,j,imax;

int v[10001];
long s,smax;
main()
{ f>>n>>k;
```

Varianta PASCAL

```
type vect=array[1..2000] of integer;
var f:text;
n,k,max,min,i,j,s,p:integer;

begin
assign(f,'bac.txt');reset(f);
read(f,n,k);
```

```

for(i=1;i<=k;i++)
{f>>v[i]; S=S+v[i];}
imax=1;i=1;
for(j=k+1;j<=n;j++)
{f>>v[j];S=S-v[i];S=S+v[j];i++;
if (S>Smax){Smax=S;imax=i;}
}
cout<<imax;
f.close();
}

```

```

s:=0;j:=1;
for i:=1 to n-k do
begin
read(f,v[i]);s:=s+v[i];
end;
ma:=s/k;
for(p=i<p<=n;p++)
{if (ma>max) then
begin

```

Varianta <22>

SUBIECTUL I

1. b

2.

a) 15

b) 10, 15, 25

c)

Varianta C/C++

```

#include<iostream.h>
void main()
{ int n,i,d;
cin>>n;
if(n<0) n=-n;
d=1;
for(i=2;i<=n/2;i++)
if(n%i==0) d=i;
} cout<<d;

```

Varianta PASCAL

```

var n,i,d:integer;
begin
read(n);
if n<0 then n:=-n;
d:=1;
for i:=2 to n div 2 do
if n mod i=0 then d:=i;
write(d);
end.

```

d) 25 (pătrat perfect de număr prim) sau 17 (număr prim, caz în care afișează1)

SUBIECTUL II

1. b

2. a

3. 5

4. 8 pentru **intensiv** și respectiv 4 pentru **neintensiv**

5.

Varianta C/C++

```

#include<iostream.h>
void main()
{ int n,m,i,j,k,a[100][100];
cin>>m>>n;
k=m*n;
for(i=1;i<=m;i++)
for(j=1;j<=n;j++)
a[i][j]=k--;
for(i=1;i<=m;i++)
{
for(j=1;j<=n;j++)
cout<<a[i][j]<<" ";
cout<<endl;
}
}

```

Varianta PASCAL

```

var n,m,i,j,k:integer;
a:array[1..100,1..100] of integer;
begin
read(m,n);
k:=m*n;
for i:=1 to m do
for j:=1 to n do
begin
a[i,j]:=k;
k:=k-1;
end;
for i:=1 to m do
begin
for j:=1 to n do
write(a[i,j], ' ');

```

```
writeln;
end;
end.
```

SUBIECTUL III

1. ABACABA

3. Intensiv

a) Deoarece nu putem calcula factorialul unui numar mare, vom determina ordinul de multiplicitate al cifrei 5 în $n!$ (ordinul de multiplicitate al cifrei 2 este mai mare decat al cifrei 5)

Varianta C/C++

```
int nz(int n)
{ int p=5, o=0;
  while(p<=n)
  { o=o+n/p;
    p=p*5;
  }
  return o;
}
```

Neintensiv

a) Deoarece $n \leq 12$ putem calcula factorialul

Varianta C/C++

```
int nz(int n)
{ long f=1, i, o=0;
  for(i=1; i<=n; i++)
    f=f*i;
  while(f%10==0) { f=f/10; o++; }
  return o;
}
```

b)

Varianta C/C++

```
#include<fstream.h>
int nz(int n);
void main()
{ int k,n=1;
  cin>>k;
  while(nz(n)<k) n++;
  cout<<n;
}
```

Varianta PASCAL

```
function nz(n:integer):integer;
var p,o:integer;
begin
  p:=5; o:=0;
  while p<=n do begin
    o:=o+n div p;
    p:=p*5;
  end;
  nz:=o;
end;
```

Varianta PASCAL

```
function nz(n:integer):integer;
var i,o:integer;
    f:longint;
begin
  f:=1; o:=0;
  for i:=1 to n do f:=f*i;
  while f mod 10=0 do begin
    f:=f div 10;
    o:=o+1;
  end;
  nz:=o;
end;
```

Varianta PASCAL

```
var n,k:integer;
function nz(n:integer):integer;
var p,o:integer;
begin
  p:=5; o:=0;
  while p<=n do begin
    o:=o+n div p;
    p:=p*5;
  end;
  nz:=o;
end;
begin
  n:=1;
  read(k);
  while nz(n)<k do n:=n+1;
  write(n);
end.
```


4. Explicații

Folosim o funcție care calculează cea mai mare putere a lui 2 mai mică decât parametrul **b**, iar dacă această valoare e mai mare decât **a**, atunci ea este valoarea căutată din intervalul **[a,b]**.

Varianta C/C++

```
#include<fstream.h>
int p2(int b)
{ int p=1;
  while(p<=b) p=p*2;
  return p/2;
}
void main()
{ int n,a,b;
  ifstream f("bac.txt");
  f>>n;
  for(int i=1;i<=n;i++)
  { f>>a>>b;

    if (p2(b)>=a) cout<<p2(b)<<" ";
    else cout<<0<<" ";
  }
}
```

Varianta PASCAL

```
var n,a,b,i:integer;
f:text;
function p2(b:integer):integer;
var p:integer;
begin
  p:=1;
  while p<=b do p:=p*2;
  p2:=p div 2;
end;
begin
  assign(f,'bac.txt');
  reset(f);

  readln(f,n);
  for i:=1 to n do begin
    readln(f,a,b);
    if p2(b)>=a then write(p2(b),'
')
    else write(0,' ');
  end;
end.
```

Varianta <23 >

SUBIECTUL I

1. a

a) 4

b) 4, 14, 9

c)

Varianta C/C++

```
#include <iostream.h>
void main()
{
  int a,b,p;
  cin>>a>>b;
  p=0;
  while(a!=b)
  {
    p=p+1;
    if(a<b) a=a+2;
    else b=b+3;
  }
  cout<<p;
}
```

d)

citește a, b (numere întregi)

p←0

dacă a≠b atunci

```
| execută
| | p← p+1
| | dacă a<b atunci
| | | a←a+2
```

Varianta PASCAL

```
var a,b,p:integer;
begin
  readln(a,b);;
  p:=0;
  while a<>b do
  begin
    p:=p+1;
    if a<b then a:=a+2
    else b:=b+3;
  end;
  writeln(p);
end.
```

```
| | |altfel
| | | b←b+3
| | ■
| cât timp a≠b
```

■
scrie p

SUBIECTUL II

1. b

2. abs

3. 2, 3 pentru **intensiv**, respectiv 5 pentru **neintensiv**

4. 300 pentru **intensiv**, respectiv 200 pentru **neintensiv**

5.

Varianta C/C++

```
#include <iostream.h>
int a[101][101],m,n;
void main()
{
    cin>>m>>n;
    for(int i=1;i<=m;i++)
        a[i][1]=i;
    for(int j=1;j<=n;j++)
        a[1][j]=j;
    for(i=2;i<=m;i++)
        for(j=2;j<=n;j++)
            a[i][j]=a[i-1][j]+a[i][j-1];
    cout<<a[m][n];
}
```

Varianta PASCAL

```
var a:array[1..100,1..10] of
integer;
m,n,i,j:integer;
begin
    readln(m,n);
    for i:=1 to m do
        a[i,1]:=i;
    for j:=1 to n do
        a[1,j]:=j;
    for i:=2 to m do
        for j:=2 to n do
            a[i,j]:=a[i-1,j]+a[i,j-1];
    writeln(a[m,n]);
end.
```

SUBIECTUL III

1. d

2. 9

3.

a)

Varianta C/C++

```
void shift(int n, int x[100])
{
    int aux;
    aux = x[1];
    for(int i=1;i<=n-1;i++)
        x[i]=x[i+1];
    x[n]=aux;
}
```

Varianta PASCAL

```
procedure shift(n:integer;var
x:vector);
var aux,i:integer;
begin
    aux:=x[1];
    for i:=1 to n-1 do
        x[i]:=x[i+1];
    x[n]:=aux;
end;
```

b) Permutăm circular spre stânga cele **n** elemente, apoi primele **n+1, n+2...** până la primele 2.

Pentru exemplul dat, se obțin pe rând configurațiile:

1 2 3 4 5 – inițial

2 3 4 5 1

3 4 5 2 1

4 5 3 2 1

5 4 3 2 1

Varianta C/C++

```
#include <iostream.h>
```

Varianta PASCAL

```
type vector=array[1..100] of integer;
```

```
void shift(int n, int x[100])
{
    int aux;
    aux = x[1];
    for(int i=1;i<=n-1;i++)
        x[i]=x[i+1];
    x[n]=aux;
}
void main()
{
    int n, x[100];
    cin>>n;
    for(int i=1;i<=n;i++)
        cin>>x[i];
    for(i=n;i>=2;i--)
        shift(i,x);
    for(i=1;i<=n;i++)
        cout<<x[i]<<" ";
}
```

4. Intensiv

Am folosit o funcție care returnează 1 dac acele două intervale transmise ca parametri sunt disjuncte și 0 în caz contrar. În **main** am verificat pentru fiecare interval cu câte intervale este disjunct, iar dac este disjunct cu **n-1** intervale, atunci îl afișăm.

Varianta C/C++

```
#include <fstream.h>

struct interval{
    int a, b;
};

int n;
interval x[1000];

int check(interval x, interval y)
{
    return x.b<y.a||y.b<x.a;
}

void main()
{
    ifstream fin ("bac.txt");
    fin>>n;

    for(int i=1;i<=n;i++)
        fin>>x[i].a>>x[i].b;
    for(i=1;i<=n;i++)
    {
        int p=0;
        for(int j= 1; j <=n; j++)
            if(check(x[i],x[j])) p++;
        if(p==n-1) cout<<x[i].a<<"
"<<x[i].b<<endl;
    }
}
```

Neintensiv

```
#include <fstream.h>

struct interval{
    int a, b;
```

```
var n,i:integer;
    x:vector;
procedure shift(n:integer;var
x:vector);
var aux,i:integer;
begin
    aux:=x[1];
    for i:=1 to n-1 do
        x[i]:=x[i+1];
    x[n]:=aux;
end;

begin
    readln(n);
    for i:=1 to n do read(x[i]);
    for i:=n downto 2 do
        shift(i,x);
    for i:=1 to n do
        end. write(x[i], ' ');
end.
```

Varianta PASCAL

```
type interval=record
    a, b:integer;
end;

var n,i,j,p:integer;
x:array[1..1000] of interval;
f:text;
function check(x,y:interval):integer;
begin
    if (x.b<y.a) or (y.b<x.a) then
        check:=1
    else check:=0;
end;

begin
    assign(f,'bac.txt');
    reset(f);
    readln(f,n);

    for i:=1 to n do
        readln(f,x[i].a,x[i].b);
    for i:=1 to n do begin
        p:=0;
        for j:=1 to n do
            if check(x[i],x[j])=1 then
                p:=p+1;
        if p=n-1 then writeln(x[i].a,'
',x[i].b);
    end;
end.

type interval=record
    a, b:integer;
end;
```

```
};

int n;
interval x[1000];

void main()
{int max;
 interval imax;
 ifstream fin ("bac.txt");
 fin>>n;
 for(int i=1;i<=n;i++)
     fin>>x[i].a>>x[i].b;
 max=x[1].b-x[1].a;
 imax=x[1];
 for(i=2;i<=n;i++)
     if(x[i].b-x[i].a>max)
         { max=x[i].b-x[i].a;
           imax=x[i];

         }
     else
         if(x[i].b-x[i].a==max)
             if(x[i].b<imax.b)
                 { max=x[i].b-x[i].a;
                   imax=x[i];
                 }
 cout<<imax.a<<" "<<imax.b;
 }
```

```
var n,i,j,max:integer;
x:array[1..1000] of interval;
f:text;
imax:integer;
begin
 assign(f,'bac.txt');
 reset(f);
 readln(f,n);
 for i:=1 to n do
     readln(f,x[i].a,x[i].b);
 max:=x[1].b-x[1].a;
 imax:=x[1];
 for i:=2 to n do
     if x[i].b-x[i].a>max then begin
         max:=x[i].b-x[i].a;
         imax:=x[i];
     end
 else
     if x[i].b-x[i].a==max then begin
         if x[i].b<imax.b then
             max:=x[i].b-x[i].a;
             imax:=x[i];
         end;
     end;
 writeln(imax.a,' ',imax.b);
end.
```

Varianta <24>

SUBIECTUL I

1. d

2.

a) 75

b) 60

c)

Varianta C/C++

```
#include <iostream.h>
```

```
int a,b,p,q;
void main()
{
 cin>>a>>b;
 p=a;
 q=b;
 if(p==0||q==0)
 {
     p=p*q;
     q=p*q;
 }
 while(p!=q)
     if(p<q) p=p+a;
     else q=q+b;
 cout<<p;
 }
```

d)

citește a, b (numere naturale)

p←a

Varianta PASCAL

```
var a,b,p,q:integer;
begin
 readln(a,b);
 p:=a;
 q:=b;
 if (p=0) and (q=0) then
     begin
         p:=p*q;
         q:=p*q;
     end;
 while p<>q do
     if p<q then p:=p+a;
     else q:=q+b;
     writeln(p);
end.
```

```

q←b
dacă p=0 sau q=0
|   p←p*q
|   q←p*q
■
dacă p≠q atunci
|   execută
|   |   dacă p<q atunci
|   |   |   p←p+a
|   |   |altfel
|   |   |   q←q+b
|   |   ■
|   cât timp p≠q
■
scrie p
    
```

SUBIECTUL II

1. a
2. b
- 3.

Varianta C/C++

```

strcpy(s+1,s+2); - intensiv
strcpy(s,s+2); - neintensiv
    
```

Varianta PASCAL

```

delete(s,2,1); - intensiv
delete(s,1,2); - neintensiv
    
```

4. O astfel de matrice se poate obține prin permutări circulare pornind de la prima linie completată cu valorile 1 2 3 ...n. Fiecare linie, începând cu a doua, se obține ca permutare circulară a liniei anterioare.

Altfel, putem folosi o formulă cu care să calculăm direct valoarea fiecărui element din

matrice în funcție de indicii săi.

```

Varianta C/C++
for(i=1;i<=n;i++) a[1][i]=i;
for(i=2;i<=n;i++)
{ for(j=1;j<=n-1;j++)
  a[i][j]=a[i-1][j+1];
  a[i][n]=a[i-1][1];
}
    
```

Sau

```

for(i=0;i<n;i++)
  for(j=0;j<n;j++)
    a[i][j]=(i+j)%5+1;
    
```

Varianta PASCAL

```

for i:=1 to n do a[1,i]:=i;
for i:=2 to n do
begin
  for j:=1 to n-1 do
    a[i,j]:=a[i-1,j+1];
  a[i,n]:=a[i-1,1];
end;
    
```

sau

```

for i:=0 to n do
  for j:=1 to n do
    a[i,j]:=(i+j-2) mod 5+1;
    
```

5. Explicații

Parcurgem în paralel cele două șiruri de la sfârșit spre început până când găsim două litere diferite în cele două șiruri. Sufixul căutat începe la poziția următoare poziției la care s-au găsit cele două litere diferite.

Varianta C/C++

```

#include <iostream.h>
#include <string.h>
void main()
{ char a[100], b[100];

  int i,j;
  cin>>a>>b;
  i=strlen(a)-1;
  j=strlen(b)-1;
    
```

Varianta PASCAL

```

var a,b:string;
i,j:integer;
begin
  readln(a);

  readln(b);
  i:=length(a);
  j:=length(b);
  while a[i]=b[j] do
    
```

```
while (a[i]==b[j])
{ i--;
  j--;
}
cout<<a+strlen(a)-i;
}
```

```
begin
  i:=i-1;
  j:=j-1;
end;
for j:=i+1 to length(a) do
  write(a[j]);
end.
```

SUBIECTUL III

1. a

2. 3, 2

3.

a)

Varianta C/C++

```
void p(int n, int x[101], int &mini,
int &maxi, int &sum)
{ int i;
  mini=maxi=sum=x[1];
  for(i=2;i<=n;i++)
  { sum=sum+x[i];
    if(x[i]>maxi) maxi=x[i];
    if(x[i]<mini) mini=x[i];
  }
}
```

Varianta PASCAL

```
procedure p(n:integer; x:sir; var
mini,maxi,sum:integer);
var i:integer;
begin
  mini:=x[1]; maxi:=x[1];sum:=x[1];
  for i:=2 to n do
  begin
    sum:=sum+x[i];
    if x[i]>maxi then maxi:=x[i];
    if x[i]<mini then mini:=x[i];
  end;
end;
```

b)

Varianta C/C++

```
#include <iostream.h>
#include <iomanip.h>
void p(int n, int x[101], int &mini,
int &maxi, int &sum)
{ int i;
  mini=maxi=sum=x[1];
  for(i=2;i<=n;i++)
  { sum=sum+x[i];
    if(x[i]>maxi) maxi=x[i];
    if(x[i]<mini) mini=x[i];
  }
}
```

Varianta PASCAL

```
type sir=array[1..100] of integer;
var n,i,min,max,s:integer;
x:sir;
ma:real;
procedure p(n:integer; x:sir; var
mini,maxi,sum:integer);
var i:integer;
begin
  mini:=x[1]; maxi:=x[1];sum:=x[1];
  for i:=2 to n do
  begin
    sum:=sum+x[i];
    if x[i]>maxi then maxi:=x[i];
    if x[i]<mini then mini:=x[i];
  end;
end;
```

```
void main()
{ int n, x[101], i, min, max, s;
  cin>>n;
  for(i=1;i<=n;i++) cin>>x[i];
  p(n, x, min, max, s);
  float ma=(float)(s-min-max)/(n-2);
  cout<<setprecision(3)<<ma;
}
```

```
begin
  readln(n);
  for i:=1 to n do readln(x[i]);
  p(n, x, min, max, s);
  ma:=(s-min-max)/(n-2);
  write(ma:7:3);
end.
```

4. a)

Intensiv

Varianta C/C++

```
#include <fstream.h>
```

Varianta PASCAL

Program

```
void main()
{ int n, x[30001],i,a,b,k;
  ifstream f("bac.txt");
  f>>n;
  for(i=1;i<=n;i++) f>>x[i];
  f>>a>>b;
  if(a<x[n])
  {
    k=1;
    while(x[k]<a && x[k]<=b && k<=n)
k++;
    if(k<=n&&x[k]<=b) cout<<x[k];
    else cout<<"NU";
  }
  else cout<<"NU";
}
```

```
Var
v:array[1..100] of integer;
n,a,b:integer;

procedure afisare(x:integer)
begin
  if (x>n) then
    write('NU')
  else
    if (v[x]>b) then
      write('NU')
    else
      begin
        if(v[x]>a&&v[x]<b)
          write(v[x],' ');
        else
          afisare(x+1);
      end;
  end;

end;

begin
assign(f,'bac.txt');
reset(f);
readln(f,n);
for i:=1 to n do
  readln(f,a[i],' ');
readln(f,a,b,' ');
close(f);
afisare(1);
readln;
end.
```

b) Deoarece intervalul are mai puține elemente decât tabloul și e mai ușor să determinăm dacă un număr se găsește într-un interval, vom căuta în interval elementele din tablou. Parcurgem tabloul doar până la găsirea primului număr care se află în intervalul dat, dacă găsim un astfel de număr. Căutarea se oprește și dacă ajungem la un număr mai mare decât **b**. Dacă **a** este mai mare decât cel mai mare element din vector, nu are rost să căutăm.

Neintensiv

Varianta C/C++

```
#include <fstream.h>
void main()
{ int n, x[10001],i,a,b,k;
  ifstream f("bac.txt");
  f>>n;
  for(i=1;i<=n;i++) f>>x[i];
  f>>a>>b;
  int min=10000;
  for(i=1;i<=n;i++)
    if(x[i]<=b && x[i]>=a)
      if(x[i]<min) min=x[i];
  if(min!=10000) cout<<min;
  else cout<<"NU";
}
```

Varianta PASCAL

```
type sir=array[1..30000] of integer;
var n,i,a,b,k:integer;
    x:sir;
    f:text;
begin
  assign(f,'bac.txt');
  reset(f);
  readln(f,n);
  for i:=1 to n do readln(f,x[i]);
  readln(f,a,b);
  if a<x[n] then
    begin
      k:=1;
      while (x[k]<a) and (x[k]<=b) and
(k<=n) do k:=k+1;
      if(k<=n) and (x[k]<=b) then
        writeln(x[k])
      else writeln('NU');
    end;
```

```

end
else writeln('NU');
end.

```

Varianta <25>

SUBIECTUL I

1. c

2.

a) 12, 18

b)

citește a, b, c (numere naturale nenule)

dacă a>b

```

|   t←a
|   a←b
|   b←t

```

■

execută dacă c|a atunci

```

|   |   scrie a
|   ■
|   a←a+1

```

cât timp a<=b

c)

Varianta C/C++

```

#include<iostream.h>
void main()
{ int a,b,c,t;
  cin>>a>>b>>c;

  if(a>b)
  { t=a;
    a=b;
    b=t;
  }
  while(a<=b)
  { if(a%c==0) cout<<a;
    a++;
  }
}

```

Varianta PASCAL

```

var a,b,c,t:integer;
begin
  readln(a,b,c);
  if a>b then
  begin
    a:=b;
    b:=t;
  end;
  while a<=b do
  begin
    if a mod c=0 then writeln(a);
    a:=a+1;
  end;
end.

```

d)

$[(b+1-a)/c]$, iar dacă a și b se divid cu c se adaugă +1 la formulă

SUBIECTUL II

1. c

2. d pentru **intensiv** , iar pentru **neintensiv** răspunsul este 2, care nu se găsește între variante, sau d) cu modificarea nici o variantă din a) b) c)

3.

Varianta C/C++

```

z=sqrt(x)+(float)1/x+abs(x)

```

Varianta PASCAL

```

z:=sqrt(x)+ 1/x+abs(x)

```

4.

Intensiv

Varianta C/C++

Varianta PASCAL


```
int p=1;
for(i=0;i<=strlen(s)/2;i++)
    if(s[i]!=s[strlen(s)-1-i]) p=0;
if(p) cout<<"Corect";
else cout<<"Incorect";
```

```
p:=1;
for i:=1 to length(s) div 2 do
    if s[i]<>s[length(s)+1-i] then
p:=0;
if p=1 then write('Corect')
else write('Incorect');
```

Neintensiv - ar trebui precizat dacă șirul conține cel puțin o cifră, sau este format doar din cifre.

Varianta C/C++

Pentru cel puțin un caracter cifră :

```
int p=0;
for(i=0;i<strlen(s);i++)
    if(s[i]>='0' && s[i]<='9') p=1;
if(p) cout<<"Corect";
else cout<<"Incorect";
```

Pentru a verifica dacă toate caracterele sunt cifre :

```
int p=1;
for(i=0;i<strlen(s);i++)
    if(s[i]<'0' || s[i]>='9') p=0;
if(p) cout<<"Corect";
else cout<<"Incorect";
```

Varianta PASCAL

```
p:=0;
for i:=1 to length(s) do
    if(s[i]>='0') and (s[i]<='9')
then p:=1;
if p=1 then write('Corect')
else write('Incorect');
```

Pentru a verifica dacă toate caracterele sunt cifre :

```
p:=1;
for i:=1 to length(s) do
    if(s[i]<'0') or (s[i]>'9') then
p:=0;
if p=1 then write('Corect')
else write('Incorect');
```

5. Explicații

Deoarece 2 la puterea 100 nu poate fi reprezentat ca număr **long**, am scris o funcție care returnează:

- 0 dacă există acel pu țin o valoare 0 pe coloană, adică produsul va fi 0;

- 1 dacă toate elementele sunt egale cu 1, adică produsul este 1;
- numărul de cifre 2 plus 1, ca să facem distincție între coloanele cu un singur 2 și cele doar cu cifre de 1.

Astfel, produsul maxim îl vor avea coloanele cu număr maxim de cifre 2.

Varianta C/C++

```
#include<iostream.h>
int m,n,a[101][101];
int p2(int j)
{ int i,p=0;
  for(i=1;i<=m;i++)
    if(a[i][j]==2) p++;
    else if(a[i][j]==0) return 0;
  if(p==0) return 1;
  else return p+1;
}

void main()
{ int i,j,max;
  cin>>m>>n;
  for(i=1;i<=m;i++)
    for(j=1;j<=n;j++)
      cin>>a[i][j];
  max=0;
  for(j=1;j<=n;j++)
    if(p2(j)>=max) max=p2(j);
  for(j=1;j<=n;j++)
    if(p2(j)==max) cout<<j<<" ";
}
```

Varianta PASCAL

```
type mat=array[1..100,1..100] of
0..2;
var a:mat;
    i,j,m,n,max:integer;
function p2(j:integer):integer;
var i,p:integer;
begin
  p:=0;
  for i:=1 to m do
    if a[i,j]=2 then p:=p+1
    else if a[i,j]=0 then begin
p2:=0; exit; end;
    if p=0 then p2:=1
    else p2:=p+1;
end;

begin
  readln(m,n);
  for i:=1 to m do
    for j:=1 to n do
      read(a[i,j]);
  max:=0;
```

```

for j:=1 to n do
  if p2(j)>max then max:=p2(j);
for j:=1 to n do
  if p2(j)=max then write(j, ' ');
end.

```

SUBIECTUL III

1.c

2. 0,95

3.

Intensiv

Varianta C/C++

```

void f(int n, int a[9], int &k)
{ int i,p;
  k=0;p=0;
  for(i=n-1;i>=0;i--)
    if(a[i]%2==0) {
      k=k*10+a[i];
      p++;
    }
  if(p==0) k=-1;
}

```

Neintensiv

Varianta C/C++

```

void f(int n, int a[9])
{ int mi=-1,mp=32000,i,aux,imi,imp;
  for(i=0;i<n;i++)
    { if(a[i]%2==0)
      if(a[i]>mp) { mp=a[i];
        imp=i;}
      if(a[i]%2!=0)
        if(a[i]<mi) { mi=a[i];
          imi=i;}
    }
  if(mi!=-1 && mp!=32000)
    { aux=a[imi];
      a[imi]=a[imp];
      a[imp]=aux;
    }
}

```

4.

a)

Varianta C/C++

```

#include<iostream.h>

int cmmdc(int a, int b)
{
  int r;

```

Varianta PASCAL

```

type sir = array [1..10] of integer;
procedure f(n:integer; a:sir; var
k:integer);
var i,p:integer;
begin
  k:=0; p:=0;
  for i:=n-1 downto 0 do
    if a[i] mod 2 = 0 then
      k:=k*10+a[i];
      if p=0 then k:=-1;
  f:=k;
end;

```

Varianta PASCAL

```

type sir = array [1..10] of integer;
procedure f(n:integer; a:sir);
var mi,mp,i,aux,imi,imp:integer;
begin
  mi:=-1;
  mp:=32000;
  for i:=0 to n-1 do
    begin
      if a[i] mod 2 = 0 then
        if a[i] > mp then begin
          mp:=a[i]; imp:=i; end;
        if a[i] mod 2 <> 0 then
          if a[i] < mi then begin
            mi:=a[i]; imi:=i; end;
    end;
  if((mi <> -1) and (mp <> 32000))
  then
  begin
    aux:=a[imi];
    a[imi]:=a[imp];
    a[imp]:=aux;
  end;
end;

```

Varianta PASCAL

```

function cmmdc(a,b:integer)
:integer;
var r:integer;
begin
  while a mod b <>0 do

```

<pre> while(a%b) { r=a%b; a=b; b=r; } return r; } void main() { double x; long a,b,c; cin>>x; b=1; while(x!=(long)x) { x=x*10; } b=b*10; a=(long)x; c=cmmdc(a,b); cout<<a/c<<" "<<b/c; } </pre>	<pre> begin r:=a mod b; a:=b; b:=r; end; cmmdc:=r; end; var x:real; a,b,c:longint; begin read(x); b:=1; while x <> trunc(x) do begin x:=x*10; b:=b*10; end; c:=cmmdc(a,b); write(a/c, ' ',b/c); end. </pre>
---	--

b) Calculăm câte cifre zecimale are numărul real și construim o fracție cu numitorul 10 la puterea câte cifre zecimale are numărul real, iar la numărător este numărul real înmulțit cu 10 la puterea câte cifre zecimale. Apoi simplificăm fracția și afișăm numărătorul și numitorul în urma simplificării. Astfel, numărul de repetări maxim 7 pentru calculul puterii lui 10 și maxim 10 la 7 pentru CMMDC dacă numerele sunt prime între ele.

Varianta <26>

SUBIECTUL I

1. c

2.

a) 1 2 3 4 5 6 7 8 9 0 1

b)

Varianta C/C++

```

#include<iostream.h>
void main()
{
    int n,c,i;
    cin>>n;
    c=0;
    for(i=1;i<=n;i++)
        { c=(c+1)%10;
          cout<<c;
        }
}
                
```

c)

```

citește n (numar natural)
c←0
i←1

cât timp i<=n executa
|   c=(c+1)%10
|   scrie c
|   i←i+1
                
```

Varianta PASCAL

```

var n,c,i:integer;
begin
    read(n);
    c:=0;
    for i:=1 to n do
    begin
        c:=(c+1) mod 10;
        write(c, ' ');
    end;
end.
                
```

■

d) 10

SUBIECTUL II

1. d

2. c

3. roton

4. 4 (20, 16, 9, 8) pentru **intensiv**, 4 pentru **neintensiv**

5. Explicații

Am folosit o funcție care primește parametrii **i** și **j** și calculează produsul elementelor de pe coloana **j** exceptând elementul de pe linia **i**.

Varianta C/C++

```
#include<iostream.h>
int n,a[7][7];
int p(int i, int j)
{ int k,p=1;
  for(k=1;k<=n;k++)
    if(k!=i) p=p*a[k][j];
  return p;
}

void main()
{ int i,j;
  cin>>n;
  for(i=1;i<=n;i++)
    for(j=1;j<=n;j++)
      cin>>a[i][j];

  for(i=1;i<=n;i++)
    for(j=1;j<=n;j++)
      if(a[i][j]==p(i,j))
        cout<<a[i][j]<<" ";
}
```

Varianta PASCAL

```
type matrix=array [1..7,1..7] of
integer;

var n:integer;
    a:matrix;

function
p(i:integer;j:integer):integer;
var k,pl:integer;
begin
  pl:=1;
  for k:=1 to n do
    if k <> i then pl:=pl*a[k][j];
  p:=pl;
end;

var i,j:integer;
begin
  read(n);
  for i:=1 to n do
    for j:=1 to n do
      read(a[i,j]);
  for i:=1 to n do
    for j:=1 to n do
      if a[i,j] = p(i,j) then
        write(a[i,j], ' ');
end.
```

SUBIECTUL III

1. b

2. 8 4 2 1 1 2 1 1 pentru **intensiv**, respectiv 4 3 2 1 0 1 2 3 4 pentru **neintensiv**

3.

Varianta C/C++

```
#include<iostream.h>
void main()
{ int n,k,i;
  cin>>n>>k;
  for(i=k;i>=1;i--)
    cout<<n*i<<" ";
}
```

4.

Intensiv

Varianta PASCAL

```
var n,k,i:integer;
begin
  read(n,k);
  for i:=k downto 1 do
    write(n*i, ' ');
end.
```

a) Ștergem fiecare element de pe poziția k între i și j prin deplasarea spre stânga cu câte o poziție a elementelor de pe pozițiile $k+1, k+2, \dots, n$, după care îl ștergem pe n cu 1 pentru fiecare element șters.

Varianta C/C++

```
void sterge(int v[1000], int &n, int
i, int j)
{ int k,l;
  for(k=i;k<=j;k++)
    { for(l=i;l<n;l++) v[l]=v[l+1];
      n--;
    }
}
```

b)

Pentru fiecare grup de k elemente alăturate identice le ștergem pe primele $k-1$ folosind funcția scrisă anterior.

Varianta C/C++

```
#include<fstream.h>
void sterge(int v[1000], int &n, int
i, int j)
{ int k,l;
  for(k=i;k<=j;k++)
    { for(l=i;l<n;l++) v[l]=v[l+1];
      n--;
    }
}

void afis(int v[1000], int n)
{ int i;
  for(i=1;i<=n;i++) cout<<v[i]<<" ";
}

void main()
{ int n,v[1000],i,j;
  ifstream f("numere.in");
  f>>n;
  for(i=1;i<=n;i++) f>>v[i];
  for(i=1;i<n;i++)
    if(v[i]==v[i+1])
      { j=i;
        while(v[j]==v[j+1]&& j<n) j++;
        sterge(v,n,i,j-1);
      }
  cout<<endl;
  afis(v,n);
}
```

Varianta PASCAL

```
type sir=array [1..1000] of
integer;
var k,l:integer;

procedure sterge(v:sir;var
n:integer;i:integer;j:integer);
begin
  for k:=i to j do
    begin
      for l:=i to n-1 do
        v[l]:=v[l+1];
      n:=n-1;
    end;
end;
```

Varianta PASCAL

```
type sir=array [1..1000] of
integer;
var k,l:integer;
    f,g:text;

procedure sterge(var v:sir;var
n:integer;i:integer;j:integer);
begin
  for k:=i to j do
    begin
      for l:=i to n-1 do
        v[l]:=v[l+1];
      n:=n-1;
    end;
end;

procedure afis(v:sir;n:integer);
var i:integer;
begin
  for i:=1 to n do
    write(v[i],' ');
end;

var n,i,j:integer;
    v:sir;
begin
  assign(f,'numere.in');
  reset(f);
  read(f,n);
  for i:=1 to n do
    read(f,v[i]);

  for i:=1 to n-1 do
    if v[i] = v[i+1] then
      begin
```

```

        j:=i;
        while((v[j]=v[j+1]) and
(j<n)) do
            j:=j+1;
            sterge(v,n,i,j-1);
        end;

        writeln;
        afis(v,n);
        close(f);
    end.

```

Neintensiv

a) Ștergem elementul de pe poziția i prin deplasarea spre stânga cu o poziție a elementelor de pe pozițiile $i+1, i+2, \dots, n$, după care îl ștergem pe cel de pe poziția i .

Varianta C/C++

```

void sterge(int v[1000], int &n, int
i)
{ int l;
  for(l=i;l<n;l++) v[l]=v[l+1];
  n--;
}

```

Varianta PASCAL

```

procedure sterge(var v:sir;var
n:integer;i:integer);
var l:integer;
begin
  for l:=i to n-1 do v[l]:=v[l+1];
  n:=n-1;
end;

```

b)

Pentru fiecare 2 elemente alăturate identice îl ștergem pe cel de pe poziția i folosind funcția scrisă anterior.

Varianta C/C++

```

#include<fstream.h>
void sterge(int v[1000], int &n, int
i)
{ int l;
  for(l=i;l<n;l++) v[l]=v[l+1];
  n--;
}

void afis(int v[1000], int n)
{ int i;
  for(i=1;i<=n;i++) cout<<v[i]<<" ";
}

void main()
{ int n,v[1000],i,j;
  ifstream f("numere.in");
  f>>n;
  for(i=1;i<=n;i++) f>>v[i];
  for(i=1;i<n;i++)
    if(v[i]==v[i+1])
    {
      sterge(v,n,i);
      i--;
    }
  cout<<endl;
  afis(v,n);
}

```

Varianta PASCAL

```

type sir=array [1..1000] of
integer;

procedure sterge(var v:sir;var
n:integer;i:integer);
var l:integer;
begin
  for l:=i to n-1 do
    v[l]:=v[l+1];
  n:=n-1;
end;

procedure afis(v:sir;n:integer);
var i:integer;
begin
  for i:=1 to n do
    write(v[i], ' ');
end;

var n,i,j:integer;
v:sir;
f:text;
begin
  assign(f,'numere.in');
  reset(f);
  read(f,n);
  for i:=1 to n do
    read(f,v[i]);
  i:=1;
  while i<=n-1 do
    if v[i] = v[i+1] then

```

```

        sterge(v,n,i)
    else i:=i+1;
afis(v,n);
close(f);
end.

```

Varianta <27>

SUBIECTUL I

1. a

2.

a) 2329

b)

Varianta C/C++

```

#include<iostream.h>
void main()
{ int a,b,c,d,p;

  cin>>a>>b;
  c=0; d=0; p=1;
  while(a+b+c>0)
  { c=a%10+b%10+c;
    d=d+(c%10)*p;
    p=p*10;
    a=a/10;
    b=b/10;
    c=c/10;
  }
  cout<<d;
}

```

e) citește a, b (numere naturale)

d←0

p←1

execută

```

|   c←a%10+b%10+c
|   d←d+(c%10)*p
|   p←p*10
|   a←[a/10]
|   b←[b/10]
|   c←[c/10]

```

cât timp a+b+c>0

scrie d

d)

citește a, b (numere naturale)

d←a+b

scrie d

Varianta PASCAL

```

var a,b,c,d,p:integer;
begin
  read(a,b);

  c:=0; d:=0; p:=1;
  while((a+b+c) > 0) do
  begin
    c:=(a mod 10)+(b mod 10)+c;
    d:=d+(c mod 10)*p;
    p:=p*10;
    a:=a div 10;
    b:=b div 10;
    c:=c div 10;
  end;
  write(d);
end.

```

SUBIECTUL II

1. a

2. b

3.

Varianta C/C++

H.x=F.x*G.y+F.y*G.x;

H.y=F.y*G.y;

4. 120 pentru **intensiv**, respectiv 6 pentru **neintensiv**

5.

Varianta PASCAL

H.x:=F.x*G.y+F.y*G.x;

H.y:=F.y*G.y;

Varianta C/C++

```
#include <iostream.h>

int a[101][101],n;

void main()
{
    cin>>n;
    for(int i=1;i<=n;i++)
        a[i][1]=a[i][n]=1;
    for(int j=1;j<=n;j++)
        a[1][j]=1;
    for(i=2;i<=n;i++)
        for(j=2;j<=n-1;j++)
            a[i][j]=a[i-1][j-1]+a[i-1][j]+a[i-1][j+1];
    for(i=1;i<=n;i++)
        { for(j=1;j<=n;j++)
            cout<<a[i][j]<<" ";
        }
}
```

Varianta PASCAL

```
type matrix=array [1..101,1..101]
of integer;

var a:matrix;
    n,i,j:integer;

begin
    read(n);
    for i:=1 to n do
        begin
            a[i,1]:=1;
            a[i,n]:=1;
        end;
    for j:=1 to n do
        a[1,j]:=1;
    for i:=2 to n do
        for j:=2 to n-1 do
            a[i,j]:=a[i-1,j-1]+a[i-1,j]+a[i-1,j+1];
        end;
    for i:=1 to n do
        begin
            for j:=1 to n do
                write(a[i,j], ' ');
            writeln;
        end;
    end.
```

SUBIECTUL III

1. c

2. 1 7 2 6 3 5 4 4 4 4 4 pentru **intensiv**, respectiv 4 4 4 pentru **neintensiv**

3. Numărăm câte cifre are y , și calculăm în variabila p 10 la putere acest număr de cifre.

Numărul real căutat se construiește ca $(x \cdot p + y) / p$;

Varianta C/C++

```
float nreal( int x, int y)
{ float r;
  int p=1,a=y;
  r=x;
  while(a)
  { p=p*10;
    a=a/10;
  }
  r=r*p+y;
  r=r/p;
  return r;
}
```

Varianta PASCAL

```
function
nreal(x:integer;y:integer):real;
var r:real;
    p,a:integer;
begin
    p:=1; a:=y; r:=x;
    while(a <> 0 ) do
        begin
            p:=p*10;
            a:=a div 10;
        end;
    r:=r*p+y;
    r:=r / p;
    nreal:=r;
end;

begin
end.
```

4. Intensiv

a)

Varianta C/C++

```
#include<fstream.h>
```

Varianta PASCAL

```
function
nrint(x:real;y:real):integer;
```



```
int nrint(float x, float y)
{ int a,p=0;
  a=(int)x+1;
  while(a<y) { p++; a++; }
  return p;
}

void main()
{ ifstream f("numere.in");
  int n,min,p;
  float x,y;
  f>>n;
  f>>x>>y;
  min=nrint(x,y);
  f>>x;
  for(int i=4;i<=n;i++)
  { f>>y;
    p=nrint(x,y);
    if(p<min) min=p;

  } x=y;
  cout<<min;
  f.close();
}
```

```
var a,p:integer;
begin
  p:=0;
  a:=trunc(x)+1;
  while(a<y) do
  begin
    p:=p+1;
    a:=a+1;
  end;
  nrint:=p;
end;

var n,min,p,i:integer;
    x,y:real;
    f:text;

begin
  assign(f,'numere.in');

  reset(f);
  read(f,x,y);
  min:=nrint(x,y);
  read(f,x);
  for i:=4 to n do
  begin
    read(f,y);
    p:=nrint(x,y);
    if p < min then min:=p;
    x:=y;
  end;
  write(min);

  close(f);
end.
```

b) Găsim numărul minim de numere întregi care apar într-un interval format din două numere reale citite consecutiv. Acest număr minim ne garantează că în oricare interval format din două numere reale citite consecutiv se vor găsi cel puțin atâtea numere întregi. Funcția nrint calculează câte numere întregi există între cele două numere reale transmise prin parametri.

Numărul total de pași făcut este de fapt egal cu câte numere întregi există între cele mai mare și cel mai mic număr real citit.

Neintensiv

Varianta C/C++

```
#include<fstream.h>

void main()
{ ifstream f("numere.in");
  int n,p;
  float x,y,min,max;
  f>>n;
  f>>x;
  min=x;
  max=x;
  for(int i=2;i<=n;i++)
  { f>>y;
    if(y<min) min=y;
    else if(y>max) max=y;
    x=y;
  }
```

Varianta PASCAL

```
var n,p,i:integer;
    x,y,min,max:real;
    f:text;

begin
  assign(f,'numere.in');
  reset(f);
  read(f,n);
  read(f,x);
  min:=x;
  max:=x;
  for i:=2 to n do
  begin
    read(f,y);
    if y < min then min:=y
```

```

}
cout<<(int)min<<" "<<(int)max+1;
f.close();
}

```

```

else if y > max then max:=y;
x:=y;
end;
if max<>trunc(max) then
max:=max+1;
write(trunc(min), '
',trunc(max)); close(f);
end.

```

b) Găsim elementul maxim și elementul minim dintre elementele din fișier, iar intervalul căutat este $[[\min], [\max]+1]$ ($[x]$ este partea întreagă din x). Astfel problema se poate rezolva încăde la citire, fără a mai fi nevoie de memorarea elementelor într-un vector.

Varianta <28>

SUBIECTUL I

1. b

2.

a) 2

b)

Varianta C/C++

```

#include<iostream.h>
#include<math.h>
void main()
{ float x;
  int y;
  cin>>x;
  y=floor(x);
  x=x-y;
  while(x!=floor(x)) x=x*10;
  if(x==y) cout<<1;
  else cout<<2;
}

```

c)

```

citește x (numar real pozitiv)
y←[x]
x←x-y
dacă x≠[x] atunci
| execută
|   | x← x*10
|   cât timp x≠[x]

```

```

■ dacă x=y atunci
|   scrie 1
|   altfel
|   scrie 2

```

■

d) 12.12

SUBIECTUL II

1. b

2. c

3: 0 3 4 5 1 pentru **intensiv**, 3 pentru **neintensiv**

5. Explicații

Varianta PASCAL

```

var y:integer;
x:real;

begin
  read(x);
  y:=trunc(x);
  x:=x-y;
  while x <> trunc(x) do
x:=x*10;
  if x=y then write('1')
  else write('2');
end.

```

Funcția *afis* afișează un șir de caractere din care lipsesc caracterele identice cu parametrul *v*. Folosim această funcție pentru a afișa șirurile obținute fără vocale.

Varianta C/C++

```
#include<iostream.h>
#include<string.h>

void afis(char s[10], char v)
{ int i;
  for(i=0;i<strlen(s);i++)
    if(s[i]!=v) cout<<s[i];
  cout<<endl;
}

void main()
{ char s[10],voc[]="aeiou";
  cin>>s;
  for(int i=0;i<strlen(voc);i++)
    if(strchr(s,voc[i]))
afis(s,voc[i]);
}
```

Varianta PASCAL

```
var s,voc:string;
    i:integer;

procedure afis(s:string;v:char);
var i:integer;
begin
  for i:=1 to length(s) do
    if s[i] <> v then
write(s[i]);
  writeln;
end;

begin
  voc:='aeiou';
  read(s);
  for i:=1 to length(voc) do
    if(pos(voc[i],s) <> 0 ) then
afis(s,voc[i]);
end.
```

SUBIECTUL III

1. a

2. 5

3.

Varianta C/C++

```
#include<iostream.h>

void main()
{ int n,i,p=0;
  float a[10001],s=0;
  cin>>n;
  for(i=1;i<=n;i++)
    { cin>>a[i];
      s=s+a[i];
    }
  for(i=1;i<=n;i++)
    if(a[i]==(s-a[i])/(n-1)) p++;
  cout<<p;
}
```

Varianta PASCAL

```
type sir=array [1..10001] of
real;

var n,i,p:integer;
    a:sir;
    s:real;

begin
  p:=0;
  s:=0;
  read(n);
  for i:=1 to n do
begin
  read(a[i]);

  end;s:=s+a[i];
  for i:=1 to n do
    if(a[i] = (s-a[i])/(n-1))
then p:=p+1;
  write(p);
end.
```

4.

a)

Varianta C/C++

```
int primul(int a)
{ int i=2;

  while(a%i) i++;
  return i;
}
```

Varianta PASCAL

```
function
primul(a:integer):integer;

var i:integer;
begin
  i:=2;
  while(a mod i <>0) do i:=i+1;
```

b) Un număr „aproape prim” are proprietatea că împărțit la primul său divizor obținem număr prim. Pentru numerele prime, funcția prim returnează valoarea transmisă ca parametru. De asemenea, nu luăm în considerare numerele din fișier care sunt prime.

Varianta C/C++

```
#include<fstream.h>

int primul(int a)
{ int i=2;
  while(a%i) i++;
  return i;
}

void main()
{ int n,a,i,max=0,p;
  ifstream f("numere.in");
  f>>n;
  for(i=1;i<=n;i++)
  { f>>a;
    if(primul(a)!=a)
    {
      p=primul(a);
      if(primul(a/p)==a/p)
        if(a>max) max=a;
    }
  }
  cout<<max;
  f.close();
}
```

```
primul:=i;
end;

begin
end.
```

Varianta PASCAL

```
function
primul(a:integer):integer;
var i:integer;
begin
  i:=2;
  while(a mod i <>0) do i:=i+1;
  primul:=i;
end;

var n,i,a,max,p:integer;
    f:text;

begin
  max:=0;
  assign(f,'numere.in');
  reset(f);
  read(f,n);
  for i:=1 to n do
  begin
    read(f,a);
    if primul(a) <> a then
      begin
        p:=primul(a);
        if primul(a div p) = a
          div p then
            if a > max then
              max:=a;
            end;
          end;
        write(max);
        close(f);
      end;
    end.
```

SUBIECTUL I

1. a

2.

a) 9

b) Varianta C/C++

```
#include<iostream.h>
void main()
{
  unsigned int n,m;
  cout<<"n=";

  cin>>n;
  cout<<"m=";
  cin>>m;
  while(n<=m)
```

Varianta PASCAL

```
var n,m:word;
begin
  write('n=');
  readln(n);
  write('m=');

  readln(m);
  while(n<=m) do
  begin
    n:=n+1;
```

Varianta <29>

```

    {
        n=n+1;
        m=m-1;
    }
    while (m<n)
    {
        m=m+1;
        n=n-1;
    }
    cout<<n;
}

```

c) (8,13) ;(7,13)

d) citeste n,m(numere naturale)
 scrie [(n+m)/2]
 stop

```

        m:=m-1;
    end;
    while (m<n) do
    begin
        m:=m+1;
        n:=n-1;
    end;
    write (n);
end.

```

SUBIECTUL II

1. a

2. d

3. atac

4. 7 pentru intensiv;

5 pentru neintensiv

5.

Varianta C/C++

```

#include<iostream.h>
void main()
{
    unsigned int
    H,A[50][50],x,ok,i,j,p=1;
    do{
        cout<<"dati n: ";
        cin>>n;
        }while(n==0||n>50);
    for(i=1;i<=n;i++)
        for(j=1;j<=n;j++)
        {
            cout<<"dati A["<<i<<"]["<<j<<"]:";
            cin>>A[i][j];
        }
    for(j=2;j<=n;j++)
    {
        if(A[1][j]%A[1][1]==0)
        {
            ok=0;
            x=A[1][j]/A[1][1];
            for(i=2;i<=n;i++)
                if(A[i][j]/A[i][1]!=x)
                {
                    ok=1;
                    break;
                }
        }
    }
    if(ok==0)
    p=p*x;
    cout<<"produsul este "<<p<<endl;
}

```

Varianta PASCAL

```

var n:word;
    A: array[1..50,1..50] of word;
    x,ok,i,j,p:word;

begin
    p:=1;
    x:=1;
    repeat
        write('dati n: ');
        read(n);
        until(n>=1) and(n<=50);
    for i:=1 to n do
        for j:=1 to n do
            begin
                write('dati A['<i>','<j>']: ');
                read(A[i,j]);
            end;
        for j:=2 to n do
            begin
                if A[1,j] mod A[1,1]=0 then
                    begin
                        ok:=0;
                        x:=A[1,j] div A[1,1];
                        for i:=2 to n do
                            if (A[i,j] div
                                A[i,1])<>x then
                                begin
                                    ok:=1;
                                    break;
                                end;
                    end;
                if ok=0 then
                    begin

```

}

```
p:=p*x;
end;
end;
writeln('produsul este ',p);
end.
```

SUBIECTUL III

1. c pentru intensiv,
d pentru neintensiv

2. 123 12 1 0 1 2 3 pentru intensiv,
1 2 3 pentru neintensiv

3.

a)

Varianta C/C++

```
unsigned int multipli(unsigned int
a,unsigned int b,unsigned int c)
```

```
{unsigned int i;
i:=a;
if(i<=b)
if(i%c==0)
return 1+multipli(i+c,b,c);
else
return multipli(i+1,b,c);
else
return 0;
}
```

4.

a)Varianta C/C++

```
#include<fstream.h>
void main()
{
unsigned int n,m,i,k,s,a[100],b[100],
ok=0;
ifstream fin("numere.in");
fin>>n>>m;
for(i=1;i<=n;i++)
fin>>a[i];
for(i=1;i<=m;i++)
fin>>b[i];
k=1;
i=1;
while(i<=m)
{
s=0;
while(k<=n)
{
s=s+a[k];
if(s<b[i])
k++;
else
if(s==b[i])
{
i++;
k++;
break;
}
}
}
}
```

Varianta PASCAL

```
function multipli(a,b,c:word):word;
var i:word;
```

```
begin
i:=a;
if i<=b then
if i mod c=0 then
multipli:=1+multipli(i+c,b,c)
else
multipli:=multipli(i+1,b,c)
else
multipli:=0;
end;
```

Varianta PASCAL

```
var n,m,i,k,s,ok:word;
a:array[1..100] of word;
b:array[1..100] of word;
fin:text;
begin
ok:=0;
assign (fin,'numere.in');
reset(fin);
read(fin,n,m);
for i:=1 to n do
read(fin,a[i]);
for i:=1 to m do
read(fin,b[i]);
k:=1;
i:=1;
while(i<=m) do
begin
s:=0;
while(k<=n) do
begin
s:=s+a[k];
if(s<b[i]) then
k:=k+1
else
if(s=b[i]) then
begin
i:=i+1;
```

<pre> } else { cout<<"nu "; ok=1; break; } if(ok==1) break; } if(ok==0) cout<<"da "; fin.close(); } </pre>	<pre> k:=k+1; break; end else begin writeln('nu '); ok:=1; break; end; end; if(ok=1) then break end; if(ok=0) then writeln('da '); close(fin); end. </pre>
--	--

b) Am adunat elementele consecutive din tabloul A, cât timp suma este mai micădecât un element din tabloul B.

Dacă suma este egalăcu acel element, opera ția se continuă, dacănu se întrerupe verificarea afișându-se NU.

In final, dacătoate elementele tabloului B se pot scrie ca sum ăde elemente consecutive din A, atunci se afișeazăDA.

Varianta <30>

SUBIECTUL I

1. d

2.

a) 4061

b)

Varianta C/C++

```

#include<iostream.h>
void main()
{
    unsigned int n,m,p,c;
    cout<<"n=";
    cin>>n;
    m=0;
    p=1;
    while(n>0)
    {
        c=n%10;
        if(c>0)
            c--;
        m=m+c*p;
        p=p*10;
        n=n/10;
    }
    cout<<m;
}
    
```

c)

citeste n(numar natural)

$m \leftarrow 0$

$p \leftarrow 1$

executa

$c \leftarrow n \% 10$

Varianta PASCAL

```

var n,m,p,c:word;
begin
    write('n=');
    readln(n);
    m:=0;
    p:=1;
    while (n>0) do
    begin
        c:=n mod 10;
        if (c>0) then
            c:=c-1;
        m:=m+c*p;
        p:=p*10;
        n:=n div 10;
    end;
    write(m);
end.
    
```

```
daca c>0 atunci
    c←c-1
    sfarsit_daca
m←m+c*p
p←p*10
n←[n/10]
cat_timp(n>0)
scrie m
```

d) 3119 și 3019

SUBIECTUL II

1. c
2. a
3. 5,4,2,3
4. pe a treia pozitie pentru intensiv,

respectiv 8 pentru neintensiv

Varianta C/C++

```
#include<iostream.h>
void main()
{
    unsigned int n,i,j,A[15][15];
    do{
        cout<<"n=";
        cin>>n;
    }while(n<=2||n>15);
    for(i=1;i<=n;i++)
        for(j=2;j<=n;j++)
            A[i][j]=A[i][j-1]+A[i-1][j];
    for(i=1;i<=n;i++)
    {
        for(j=1;j<=n;j++)
            cout<<A[i][j]<<" ";
        cout<<endl;
    }
}
```

Varianta PASCAL

```
var n,i,j: word;
A:array[1..15,1..15] of word;
begin
    repeat
        write('n=');
        readln(n);
    until (n>2)and (n<=15);
    for i:=1 to n do
        begin
            A[i,1]:=1;
        end;
    for i:=2 to n do
        for j:=2 to n do
            A[i,j]:=A[i,j-1]+A[i-1,j];
    for i:=1 to n do
        begin
            for j:=1 to n do
                write(A[i,j],' ');
            writeln;
        end;
end.
```

SUBIECTUL III

1. a
2. 1 2 3 4 4 3 4 4 2 3 4 4 3 4 4 pentru **varianta intensiv**
respectiv: 4 3 2 1 pentru **neintensiv**

3.

Varianta C/C++

```
long int suma(int v[100],unsigned
int n,unsigned int i, unsigned int
j)
{
    long int s=0;
    for(unsigned int k=i;k<=j;k++)
        s=s+v[k];
    return s;
}
```

Varianta PASCAL

```
function
suma(v:vect;n,i,j:word):longint;
var s:longint;k:word;
begin
    s:=0;
    for k:=i to j do
        s:=s+v[k];
    suma:=s;
end;
```


}

end;

4

a) Varianta C/C++

Pentru intensiv

```
#include<fstream.h>
void main()
{
    ifstream fin("NUMERE.IN");
    unsigned int n,k,i,x,y;
    float nr;
    fin>>n;
    fin>>nr;
    x=(unsigned int)nr;
    y=x+1;
    k=1;
    for(i=2;i<=n;i++)
    {
        fin>>nr;
        if(nr>x && nr>y)
        {
            k++;
            x=(unsigned
int)nr;
            y=x+1;
        }
        cout<<k;
        fin.close();
    }
}
```

Varianta PASCAL

Pentru intensiv

```
var fin:text;
    var n,k,i,x,y:word; nr:real;
begin
    assign(fin,'numere.in');
    reset(fin);
    read(fin,n);
    read(fin,nr);
    x:=trunc(nr);
    y:=x+1;
    k:=1;
    for i:=2 to n do
    begin
        read(fin,nr);
        if(nr>x) and (nr>y)
    then
        begin
            k:=k+1;
            x:=trunc(nr);
            y:=x+1;
        end;
    end;
    write(' ',k);
    close(fin);
end.
```

b) Se citește primul număr, se formează intervalul [x,y], conform enunțului, apoi începând cu al doilea număr se verifică apartenența la interval.

Dacă numărul citit nu aparține intervalului anterior se formează următorul interval.

Eficiența constă în faptul că nu se folosesc tablouri și are doar o structură repetitivă.

4. Varianta neintensiv

Varianta C/C++

```
a) #include<fstream.h>
void main()
{
    ifstream fin("NUMERE.IN");
    unsigned int n,k,i;

    float nr,x;
    fin>>n;
    fin>>nr;
    k=1;
    for(i=2;i<=n;i++)
    {
        fin>>x;
        if(nr!=x )
        {
            k++;
            nr=x;
        }
    }
    cout<<k;
    fin.close();
}
```

Varianta neintensiv

Varianta PASCAL

```
var fin:text;
    var n,k,i: word; nr,x:real;
begin
    assign (fin,'numere.in');
    reset(fin);

    read(fin,n);
    read(fin,nr);
    k:=1;
    for i:=2 to n do
    begin
        read(fin,x);
        if (n<>x ) then
        begin
            k:=k+1;
            nr:=x;
        end;
    end;
    write(' ',k);
    close(fin);
end.
```

b) Se memorează primul număr din șir, se verifică fiecare element următor cu acesta și dacă nu sunt egale se schimbă valoarea lui, mărindu-se și contorul cu 1. Eficiența constă în faptul că nu se folosesc tablouri, sau subprograme și se parcurg elementele o singură dată.

Varianta <31>

SUBIECTUL I

- 1. b
- 2.
- a) 16
- b) 3 și 5
- c)

Varianta C/C++

```
#include<iostream.h>
void main()
{
    unsigned int a,k,b;
    do{
        cout<<"a=";
        cin>>a;
    }while(a==0);
    k=0;
    b=(a+1)*(a+2)/2;
    while(b>=a)
    {
        b=b-a;
        k++;
    }
    cout<<b<<" "<<k;
}
```

d)

citeste a (numar natural nenul)
 $b \leftarrow [(a+1) * (a+2) / 2]$
 scrie b mod a, b div a

Varianta PASCAL

```
var a,k,b:word;
begin
    repeat
        write('a=');
        read(a);
    until(a<>0);
    k:=0;
    b:=trunc((a+1)*(a+2)/2);
    while(b>=a) do
    begin
        b:=b-a;
        k:=k+1;
    end;
    write(b,' ',k);
end.
```

SUBIECTUL II

1. c

3. Varianta C/C++

```
struct autoturism
{
    char marca[20];
    unsigned int
    anul_fabricatiei;
}x;
```

4. 5 7 9 11 pentru intensiv,

Varianta PASCAL

```
type autoturism=record
marca:string[20];
anul_fabricatiei:word;
end;
var x:autoturism;
```

respectiv

		200
	100	100
6	6	6
11	11	11

pentru neintensiv

5. Pentru intensiv:

Varianta C/C++

```
#include<iostream.h>
#include<string.h>

struct nod{
    char info[20];
    nod *urm;
} *prim, *p;

void main()
{
    unsigned int n,i,k=0, lung;
    char cuv[20];
    struct nod * nou;
    do{
        cout<<"n=";
        cin>>n;
    }while (n>100);
    do{
        cout<<"dati un cuvant ";
        cin>>cuv;
        cin.get();
        k++;
        lung=strlen(cuv);
    }while (cuv[0]!=cuv[lung-1]&& k<n);
    if(k<=n && cuv[0]==cuv[lung-1])
    {
        prim=new nod;
        strcpy(prim->info, cuv);
        prim->urm=NULL;

        for(i=k+1; i<=n; i++)
            cout<<"dati un cuvant ";
            cin>>cuv;
            cin.get();
            lung=strlen(cuv);
            if (cuv[0]==cuv[lung-1])
            {
                nou=new nod;
                strcpy(nou->info, cuv);
                nou->urm=prim;
                prim=nou;
            }
    }
    p=prim;
    while (p!=NULL)
    {
        cout<<p->info<<" ";
        p=p->urm;
    }
}
```

5.

Pentru neintensiv:

Varianta C/C++

```
#include<iostream.h>
void main()
{
```

Varianta PASCAL

```
type pnod=^nod;
    nod=record
        info:string[20];
    end; urm: pnod;
var prim,p:pnod;
var n,i,k, lung:word;
    cuv :string[20];
var nou:pnod;
begin
    k:=0;
    repeat
        write('n=');
        read(n);
    until (n<=100);
    repeat
        write('dati un cuvant ');
        readln;
        read(cuv);
        readln;
        k:=k+1;
        lung:=length(cuv);
    until (cuv[1]=cuv[lung]) and(k<n);
    if(k<=n) and (cuv[1]=cuv[lung])
    then
    begin
        new(prim);
        prim^.info:=cuv;
        if prim^.urm=nil; do
            begin
                write('dati un cuvant ');
                read(cuv); readln;
                lung:=length(cuv);
                if (cuv[1]=cuv[lung]) then
                begin
                    new(nou);
                    nou^.info:=cuv;
                    nou^.urm:=prim;
                    prim:=nou;
                end;
            end;
        p:=prim;
        while (p<>NIL) do
            begin
                write(p^.info, ' ');
                p:=p^.urm;
            end;
        end;
        end.
```

Pentru neintensiv:

Varianta PASCAL

```
var a:array[1..11,1..8] of
integer; i,j,s:integer;
begin
```

```
int a[11][8], i, j, s=0;
for (i=1; i<=10; i++)
    for (j=1; j<=7; j++)
do {
    cout<<"a["<<i<<"] ["<<j<<"]=";
    cin>>a[i][j];
}while (a[i][j]<-999 || a[i][j]>999);
    for (i=1; i<=10; i++)
        s=s+a[i][1];
    for (i=1; i<=10; i++)
        s=s+a[i][7];
    for (j=2; j<=6; j++)
        s=s+a[1][j];
    for (j=2; j<=6; j++)
        s=s+a[10][j];
cout<<s;
}
```

```
s:=0;
for i:=1 to 10 do
    for j:=1 to 7 do
        repeat
            write('a[' , i, ']' [' , j, ']=');
            readln(a[i, j]);
        until (a[i, j]>=-999) and
(a[i, j]<=999);
        for i:=1 to 10 do
            s:=s+a[i, 1];
        for i:=1 to 10 do
            s:=s+a[i, 7];
        for j:=2 to 6 do
            s:=s+a[1, j];
        for j:=2 to 6 do
            s:=s+a[10, j];
        write(' ', s);
end.
```

SUBIECTUL III

1. c pentru varianta intensiv, respectiv d pentru neintensiv

2. 6 8 10 12 10 8 pentru varianta intensiv respectiv: 5 7 9 pentru neintensiv

3. Pentru intensiv

Varianta C/C++

```
float suma(float x[100], unsigned
int n, unsigned int m)
{
    unsigned int i, j;
    float aux, s=0;
    for (i=1; i<=n-1; i++)
        for (j=i+1; j<=n; j++)
            if (x[i]>x[j])
                {aux=x[i];
                x[i]=x[j];
                x[j]=aux;
                }
    for (i=1; i<=m; i++)
        s=s+x[i];
    return s;
}
```

3. Pentru neintensiv:

Greșeală în enunț: se compară n cu m, care nu există!

Varianta C/C++

```
void ordonare(float x[100], unsigned
int n)
{
    unsigned int i, j;
    float aux;
    for (i=1; i<=n-1; i++)
        for (j=i+1; j<=n; j++)
            if (x[i]%2==1 && x[j]%2==1)
                if (x[i]>x[j])
                    {aux=x[i];
                    x[i]=x[j];
                    x[j]=aux;
                    }
```

Varianta PASCAL

```
function suma(x:vect; n, m:word):real;
var i, j:word; aux, s:real;
begin
    s:=0;
    for i:=1 to n-1 do
        for j:=i+1 to n do
            if (x[i]>x[j]) then
                begin
                    aux:=x[i];
                    x[i]:=x[j];
                    x[j]:=aux;
                end;
        for i:=1 to m do
            s:=s+x[i];
    suma:=s;
end;
```

Pentru neintensiv:

Varianta PASCAL

```
procedure suma(x:vect; n:word);
var i, j:word; aux:real;
begin
    s:=0;
    for i:=1 to n-1 do
        for j:=i+1 to n do
            if (x[i] mod 2=1) and (x[j] mod 2=1)
then
                if (x[i]>x[j]) then
                    begin
                        aux:=x[i];
                        x[i]:=x[j];
                        x[j]:=aux;
```

```

        x[j]=aux;
    }
    for(i=1;i<=n;i++)
        cout<<x[i]<<" ";
}

```

4 a)

Varianta C/C++

```

#include<fstream.h>
void main()
{
    ifstream fin("numere.txt");
    unsigned int n,i;
    int x,y,max,min;
    fin>>n;
    fin>>x>>y;
    max=x;
    min=y;
    for(i=2;i<=n;i++)
    {
        fin>>x>>y;
        if(max<x)
            max=x;
        if(min>y)
            min=y;
    }
    if(max<min)
        cout<<max<<" "<<min;
    else
        cout<<"0";
}

```

```

        end;
    for i:=1 to n do
        write(x[i],' ');
end;

```

Varianta PASCAL

```

var fin:text;
var x,y,max,min: integer;n,i:word;
begin
    assign(fin,'numere.txt');
    reset(fin);
    read(fin,n);
    read(fin,x,y);
    max:=x;
    min:=y;
    for i:=2 to n do
    begin
        read(fin,x,y);
        if(max<x) then
            max:=x;
        if(min>y) then
            min:=y;
    end;
    if(max<min) then
        write(max,' ',min)
    else
        write('0');
end.

```

b) Se determină minimul pentru valorile lui x și maximul pentru valorile lui y , iar la final se compară aceste valori, în sensul că dacă $\max < \min$ atunci există intersecție, în caz contrar se va afișa 0.

Varianta <32>

SUBIECTUL I

1. d

2.

a) 9 18 36 72 144 288

b) 1199

Varianta C/C++

```

#include<iostream.h>
void main()
{
    unsigned int a,b,c;
    cout<<"a=";
    cin>>a;
    cout<<"b=";
    cin>>b;
    if(a>b)
    {
        c=b;
        b=a;
        a=c;
    }
}

```

Varianta PASCAL

```

var a,b,c:word;
begin
    write('a=');
    readln(a);
    write('b=');
    readln(b);
    if(a>b) then
    begin
        c:=b;
        b:=a;
        a:=c;
    end;
    while(a<=b) do

```

<pre> } while(a<=b) { cout<<a<<" "; a=a*2; } cout<<a; } </pre>		<pre> begin write(' ',a,' '); a:=a*2; end; write(a); end. </pre>
---	--	--

d)

citeste a,b(numere naturale)
 dacă a>b atunci
 c←b
 b←a
 a←c
 sfarsit_daca
 executa
 scrie a
 a←a*2
 cat_timp a<=b
 scrie a

SUBIECTUL II

1. b pentru intensiv și a pentru neintensiv

2. d pentru intensiv și a pentru neintensiv

3. 8

**4. 3 4 5 6 7 8 pentru intensiv, respectiv pentru neintensiv: toate liniile sunt identice : 9
 8 7 dacăse corectează enunțul la primul forcontorul din jîn i.**

5. Varianta C/C++

```

#include<iostream.h>
#include<string.h>
void main()
{
    char c1,c2,sir[250];
    unsigned int i;
    cout<<"dati textul ";
    cin.get(sir,250);
    cout<<"dati c1 ";
    cin>>c1;
    cout<<"dati c2 ";
    cin>>c2;
    cout<<sir<<endl;
    for(i=0;i<strlen(sir);i++)
    {
        if(sir[i]==c1)
            sir[i]=c2;
        else
            if(sir[i]==c2)
                sir[i]=c1;
    }
    cout<<sir<<endl;
}

```

Varianta PASCAL

```

var    c1,c2:char; sir:string[250];
       k,i:word;
begin
    write('dati textul ');
        readln(sir);
        write('dati c1 ');
            readln(c1);
                write('dati c2 ');
                    readln(c2);
writelnsir);
        k:=length(sir);
        for i:=1 to k do
            begin
                if(sir[i]=c1) then
                    sir[i]:=c2
                else
                    if(sir[i]=c2) then
                        sir[i]:=c1;
            end;
        writelnsir);
end.

```

SUBIECTUL III

1. c

2. 16 4 6 18 pentru **varianta intensiv** respectiv: 16 4 **pentru neintensiv**

3.

Varianta C/C++

Pentru intensiv

```
unsigned int nr_prim(unsigned int x)
{
    unsigned int ok=0,y,i;
    if(x%2==0)
        y=x+1;
    else
        y=x;
    while(ok==0)
    {
        for(i=2;i<=sqrt(y);i++)
            if(y%i==0)
                ok=1;
    }
    if(ok==1)
    {
        ok=0;
        y=y+2;
    }
    return y;
}
```

Pentru neintensiv:

Varianta C/C++

```
void prime(unsigned int x,unsigned int y)
{
    unsigned int ok,i,j,aux;
    if(x>y)
    {aux=x;
    x=y;
    y=aux ;}
    for(i=x;i<=y;i++)
    {
        Ok=0;
        for(j=2;j<=sqrt(i);j++)
            if(i%j==0)
                {ok=1;break;}
        if(ok==0)
            cout<<i<<" ";
    }
}
```

Varianta PASCAL

Pentru intensiv

```
function nr_prim(x:word):word;
var ok,y,i:word;
begin
    ok:=0;
    if(x mod 2=0) then
        y:=x+1
    else
        y:=x;
    while(ok=0) do
        begin
            for i:=2 to trunc(sqrt(y)) do
                if (y mod i=0) then
                    ok:=1;

            if (ok=1) then
                y:=y+2;
                ok:=0;
            end;
        end ;
    nr_prim:=y;
end;
```

end;

Pentru neintensiv:

Varianta PASCAL

```
procedure prime(x,y:word);
var ok,i,j,aux:word;
begin
    if (x>y) then
        begin
            aux:=x;
            x:=y;
            y:=aux;
        end;
    for i:=x to y do
        begin
            ok:=0;
            for j:=2 to trunc(sqrt(i)) do
                if (i mod j =0) then
                    begin
                        ok:=1;
                        beak;
                    end;
            if(ok=0) then
                write(i,' ');
        end;
end;
```

end;

4. Intensiv

a) Eficiența programului constă în faptul că nu se folosesc tablouri, iar rezultatul se obține

printr-o singură parcurgere a elementelor din fișier.

Mai exact se inițializează două variabile **min1** și **min2**, apoi se citesc pe rând din fișier numerele și se schimbă valorile celor două variabile atunci când se întâlnesc numere de 2

cifre.

Dacă în final s-au schimbat valorile inițiale ale lui **min1** și **min2** atunci se vor afișa noile valori.

Dacă nu se va afișa doar **min1**.

b) Varianta C/C++

```
#include<fstream.h>
void main()
{
    long int nr;
    int min1,min2;
    ifstream fin("numere.txt");
    min1=100;
    min2=100;
    while(!fin.eof())
    {
        fin>>nr;
        if((nr>=10 &&
nr<=99) || (nr>=-99 && nr<=-10))
            if(nr<min1 && nr<min2)
                {
                    min2=min1;
                    min1=nr;
                }
            else
                {
                    if(nr>min1 && nr<min2)
                        min2=nr;
                }
    }
    fin.close();
    if(min1!=100 && min2!=100)
        cout<<min1<<" "<<min2;
    else
        if(min1!=100)
            cout<<min1;
}
```

Varianta PASCAL

```
var fin:text;
    nr:longint;
    min1,min2: integer;
begin
    assign ( fin,'numere.txt');
    reset(fin);
    min1:=100;
    min2:=100;
    while(not eof(fin)) do
    begin
        read(fin,nr);
        if(((nr>=10) and
(nr<=99)) or ((nr>=-99) and (nr<=-
10))) then
            if((nr<min1) and
(nr<min2)) then
                begin
                    min2:=min1;
                    min1:=nr;
                end
            else
                begin
                    if(nr>min1) and
(nr<min2) then
                        min2:=nr;
                    end;
                end;
        close(fin);
        if(min1<>100) and (min2<>100)
            then
                write(min1,' ',min2)
            else
                if(min1<>100) then
                    write(min1);
        end.
```

4. Pentru neintensiv:

a) Eficiența programului constă în faptul că nu se folosesc tablouri, iar rezultatul se obține printr-o singură parcurgere a elementelor din fișier.

Mai exact se inițializează două variabile **max1** și **max2** cu -100, apoi se citesc pe rând din fișier numerele și se schimbă valorile celor două variabile atunci când se întâlnesc numere pare.

Dacă în final s-au schimbat valorile inițiale ale lui **max1** și **max2** atunci se vor afișa noile valori.

Dacă nu se va afișa doar **max1**.

b) Varianta C/C++

```
#include<fstream.h>
void main()
{
    long int nr;
    int max1,max2;
```

Varianta PASCAL

```
var fin:text;
    max1,max2: integer;
begin
    assign ( fin,'numere.txt');
```



```

ifstream fin("numere.txt");
max1=-100;
max2=-100;
while(!fin.eof())
{
    fin>>nr;
    if(nr%2==0)
        if(nr>max1 &&
nr>max2)
            {
                max1=max2;
                max2=nr;
            }
        else
            {
                if(nr>max1
&& nr<max2)
                    max1=nr;
            }
}
fin.close();
if(max1!=-100 && max2!=-100)
    cout<<max1<<" "<<max2;
else
    if(max1!=-100)
        cout<<max1;
}

```

```

reset(fin);
max1:=-100;
max2:=-100;
while(not eof(fin)) do
begin
read(fin, nr);
if(nr mod 2 =0) then
if((nr>max1) and (nr>max2))
then
begin
max1:=max2;
max2:=nr;
end
else
begin
if(nr>max1) and (nr<max2)
then
max1:=nr;
end;
close(fin);
if(max1<>-100) and (max2<>-100) then
write(max1, ' ', max2)
else
if(max1<>-100) then
write(max1);
end.

```

Varianta <33>

SUBIECTUL I

1. c

2.

a) 135

b) (2,4) sau (3,9) sau (4,16)

c)

Varianta C/C++

```

#include<iostream.h>
void main()
{
    unsigned int x,y,p;
    cout<<"x=";
    cin>>x;
    cout<<"y=";
    cin>>y;
    p=0;
    do{
        if(y%2!=0)
            p=p+x;
        y=y/2;
        x=x*2;
    }while(y>=1);
    cout<<p;
}

```

d)

Varianta PASCAL

```

var    x,y,p:word;
begin
    write('x=');
    read(x);
    write('y=');
    read(y);
    p:=0;
    repeat
        if(y mod 2<>0) then
            p:=p+x;
        y:=y div 2;
        x:=x*2;
    until (y<1);
    write(p);
end.

```

citeste x,y (numere naturale)
 scrie x*y

SUBIECTUL II

1. a

2. c

3. Varianta C/C++

```
struct cerc{
    int
    abscisa,ordonata;
    float raza;
};
```

4. 2 3 4

5 6 7

8 9 10

11 12 13

14 15 16

5. Intensiv:

Varianta C/C++

```
//se declară global
struct nod{
    float info;
    nod *prec,*urm;
}*prim;
//funcția cerută este:
void prelucrare()
{
    struct nod *nou,*p;
    p=prim;
    while(p->urm)
    {
        if(p->info<0)
        {
            nou=new nod;
            nou->info=0;
            nou->urm=p->urm;
            p->urm->prec=nou;
            nou->prec=p;
            p->urm=nou;
        }
        p=p->urm;
    }
    if(p->info<0)
    {
        nou=new nod;
        nou->info=0;
        nou->urm=NULL;
        p->urm=nou;
        nou->prec=p;
    }
    p=prim;
    while(p)
    {
        cout<<p->info<<" ";
```

Varianta PASCAL

```
type cerc=record
    abscisa,ordonata:integer;
    raza: real;
end;
var x:cerc;
```

Varianta PASCAL

```
type pnod=^nod;
    nod=record
        info:real;
        prec,urm: pnod;
    end;
var prim,p:pnod;

procedure prelucrare;
var nou,p:pnod;
begin
    p:=prim;
    while (p^.urm<>NIL) do
    begin
        if (p^.info<0)
        then
            begin
                new(nou);

                nou^.info:=0;
                nou^.urm:=p^.urm;
                p^.urm^.prec:=nou;

                nou^.prec:=p;
                p^.urm:=nou;
            end;
            p:=p^.urm;
        end;
        if (p^.info<0) then
            begin
                new(nou);
                nou^.info:=0;
                nou^.urm:=NIL;
                p^.urm:=nou;
```

```

        p=p->urm;
    }
}
Obs.pentru verificare se poate completa
cu:
#include<iostream.h>
void main()
{
    int n,i;
    struct nod *nou,*ultim,*p;
    cout<<"n=";
    cin>>n;
    prim=new nod;
    cout<<"dati o informatie ";
    cin>>prim->info;
    prim->urm=prim->prec=NULL;
    ultim=prim;
    for(i=2;i<=n;i++)

    {
        nou=new nod;
        cout<<"dati informatia din
nod ";

        cin>>nou->info;
        nou->urm=NULL;
        nou->prec=ultim;
        ultim->urm=nou;
        ultim=nou;
    }
    p=prim;
    while(p)
    {
        cout<<p->info<<" ";
        p=p->urm;
    }
    cout<<endl;
    prelucrare();
}

```

5. Pentru neintensiv:

Varianta C/C++

```

#include<iostream.h>
#include<string.h>
void main()
{
    char sir[255];
    unsigned int
n,i,m=0,j,v[256],aux,k=1;
    cin.get(sir,255);
    n=strlen(sir);

    for(i=0;i<sir[i++]==' ')
        k++;
    cout<<k<<endl;
}

```

```

        nou^.prec:=p;
    end;
    p:=prim;
    while(p<>NIL) do
    begin
        write(p^.info,' ');
        p:=p^.urm;
    end;
    writeln;
end;
obs. pentru verificare se poate
completa cu:
var n,i:integer;
    nou,ultim:pnod;
begin
    write('n=');
    read(n);
    new(prim);

'); write('dati o informatie
read(prim^.info);
prim^.urm:=NIL;
prim^.prec:=NIL;
ultim:=prim;
for i:=2 to n do
begin
    new(nou);
    write('dati o
informatie ');
    read(nou^.info);
    nou^.urm:=NIL;
    nou^.prec:=ultim;
    ultim^.urm:=nou;
    ultim:=nou;
end;
p:=prim;
while(p<>NIL) do
begin
    write(p^.info,' ');
    p:=p^.urm;
end;

prelucrare;
end.

```

Pentru neintensiv:

Varianta PASCAL

```

var sir:string[255];
    n,i,m,j,aux,k:word;
    v:array[1..256] of word;
begin
    m:=0;
    k:=1;
    read(sir);
    n:=length(sir);
    for i:=1 to n do
        if(sir[i]=' ')
    then
        k:=k+1;
    writeln(k);
end;

```

```

    for(i=0;i<n;i++)
        if(sir[i]!='A' &&
sir[i]!='E'&&sir[i]!='I'&&sir[i]!='O'&&
ir[i]!='U'&&sir[i]!=' ')
            v[++m]=sir[i];
    for(i=1;i<m;i++)
        for(j=i+1;j<=m;j++)
            if(v[i]>v[j])
                {
                    aux=v[i];
                    v[i]=v[j];
                    v[j]=aux;
                }
    cout<<(char)v[1]<<" ";
    for(i=2;i<=m;i++)
        if(v[i]!=v[i-1])
            cout<<(char)v[i]<<" ";
}

```

```

    for i:=1 to n do
        if(sir[i]<>'A') and
(sir[i]<>'E') and (sir[i]<>'I')
and (sir[i]<>'O')and
(sir[i]<>'U') and (sir[i]<>' ')
then
            begin
                m:=m+1;
                v[m]:=ord(sir[i]);
            end;
    for i:=1 to m do
        for j:=i+1 to m do
            if (v[i]>v[j]) then
                begin
                    aux:=v[i];
                    v[i]:=v[j];
                    v[j]:=aux;
                end;
    write(chr(v[1]),' ');
    if i<>n then
        write(chr(v[i]),' ');
    end.

```

SUBIECTUL III

1. b

2. $re(1)=10;re(14)=3$ pentru **varianta intensiv**; respectiv: 2 3 29 pentru **neintensiv**

3. Pentru intensiv

Varianta C/C++

a)

```

int max_cif(int x[100],unsigned int
n)
{
    int ok=0,max=0;
    unsigned int i=1,k;
while(i<=n)
if((x[i]<-999|| (x[i]>-100 &&
x[i]<100)||x[i]>999))
    i++;
else
    break;
    if(i!=n+1 )
        {
            max=x[i];
            ok=1;
        }
for(k=i+1;k<=n;k++)
if((x[k]>=-999 && x[k]<=-
100)|| (x[k]>=100 && x[k]<=999))
    if(max<x[k])
        max=x[k];
if(ok==0)
    return 0;
else
    return max;
}

```

Varianta PASCAL

```

type vect=array[1..100] of integer;
function max_cif( x:vect;n:word)
:integer;
    var ok,max:integer;
        i,k:word;
begin
    ok:=0;
    max:=0;
    i:=1;
while(i<=n) do
if (x[i]<-999) or ((x[i]>-100) and
(x[i]<100)) or (x[i]>999) then
    i:=i+1
else
    break;
    if i<>(n+1 ) then
        begin
            max:=x[i];
            ok:=1;
        end;
for k:=i+1 to n do
if ((x[k]>=-999) and (x[k]<=-100))
or ((x[k]>=100) and ( x[k]<=999))
thenif (max<x[k]) then
    max:=x[k];
if(ok=0) then

```

b) Varianta C/C++

```
#include<fstream.h>
int max_cif(int [], unsigned int);
void main()
{
    ifstream fin("numere.txt");
    int x[100];
    unsigned int i, n;
    fin >> n;
    for(i=1; i<=n*n; i++)
        fin >> x[i];
    n=n*n;
    cout << max_cif(x, n);
    fin.close();
}
```

c) – programul citește valoarea lui n din fișier, creează vectorul x având $n*n$ elemente și apelează funcția.

Funcția determină cel mai mare număr de trei cifre din vector și returnează această valoare funcției *main()* în cazul în care există sau valoarea 0, în caz contrar.

- eficiența, așa cum este conceput enunțul nu prea este. Programul este eficient dacă vectorul x se creează în funcția *main()* doar din acele numere din fișier care au 3 cifre, rămânând ca funcția *max_cif* să determine cel mai mare număr de 3 cifre, dacă există și dacă nu va returna valoarea 0.

Varianta optimă: (pentru intensiv)

```
#include<fstream.h>
int max_cif(int [], unsigned int);
void main()
{
    ifstream fin("numere.txt");
    int x[100], nr;
    unsigned int i, n, k=0;
    fin >> n;
    for(i=1; i<=n*n; i++)
    {
        fin >> nr;
        if((nr >= -999 && nr <= -100) || (nr >= 100 && nr <= 999))
            x[++k]=nr;
    }
    n=k;
    cout << max_cif(x, n);
    fin.close();
}
int max_cif(int x[100], unsigned int n)
{
    int max;
    unsigned int i;
    if(n)
    {
        max=x[1];
        for(i=2; i<=n; i++)
            if(x[i]>max)
                max=x[i];
    }
}
```

```
max_cif:=0
else
max_cif:=max;
end;
```

Varianta PASCAL

```
var fin:text;
var x:vect;
    i,n:word;

begin
    assign(fin, 'numere.txt');
    reset(fin);
    read(fin, n);
    for i:=1 to n*n do
        read(fin, x[i]);
    n:=n*n;
    write(' ', max_cif(x, n));
    close(fin);
end.
```

Varianta optimă: (pentru intensiv)

```
type vect=array[1..100] of integer;
function max_cif( x:vect;n:word)
:integer;
    var max:integer;
        i:word;
begin
    if(n<>0) then
        begin
            max:=x[1];
            for i:=2 to n do
                if(x[i]>max)
                    then
                        max:=x[i];
            end;
            if(n=0) then
                max_cif:=0
            else
                max_cif:=max;
        end;
end;

var fin:text;
var x:vect;
    i,n,k:word;
    nr:integer;
begin
    assign(fin, 'numere.txt');
    reset(fin);
    read(fin, n);
    k:=0;
```

```

    }
    if(n==0)
        return 0;
    else
        return max;
}

```

3. Pentru neintensiv:

Varianta C/C++

```

int impar( unsigned long int
x[100], unsigned int n)
{
    int ok=0;
    unsigned int i;
    for(i=1; i<=n; i++)
        if(x[i]%10%2==1)
            {
                ok=1;
                break;
            }
    return ok;
}

```

4. pentru neintensiv

Varianta C/C++

a)

```

#include<fstream.h>
void main()
{
    ifstream fin("numere.txt");

    unsigned n,m,nr,i,s=0;
    fin>>n>>m;
    for(i=1; i<=m; i++)
        fin>>nr;
    for(i=m+1; i<=n; i++)
    {
        fin>>nr;
        s=s+nr;
    }
    cout<<s;
    fin.close();
}

```

```

for i:=1 to n*n do
begin
    read(fin,nr);
    if((nr>=-999) and (
nr<=-100)) or((nr>=100) and (
nr<=999)) then
begin
    k:= k+1;
    x[k]:=nr;
end;

end;
n:=k;

write(' ',max_cif(x,n));
close(fin);
end.

```

Varianta PASCAL

```

type vect=array[1..100] of longint;
function impar( x:vect;n:word)
:integer;
var i:word;
ok:integer;
begin
    ok:=0;
    for i:=1 to n do
        if((x[i] mod 10) mod 2 =1)
then
begin
ok:=1;
break;
end;
impar:=ok;
end;

```

Varianta PASCAL

```

var fin:text;
n,m,nr,i,s:word;
begin
    assign(fin,'numere.txt');

    reset(fin);
    read(fin,n,m);
    s:=0;
    for i:=1 to m do
        read(fin,nr);
    for i:=m+1 to n do
begin
        read(fin,nr);
        s:=s+nr;
end;
    write(s);
    close(fin);
end.

```

b) Se declară o variabilă **s**, în care se va calcula suma cerută și o variabilă **nr** în care se citesc pe rând numerele de pe linia 2.

Se citesc din fisier **n,m** și primele **m** numere apoi începând cu **m+1**, se adună numerele

rămase (care sunt și cele cerute).

Programul este eficient deoarece se parcurg o singură dată numerele din fișier și nu se folosesc tablouri.

SUBIECTUL I

1. a

2.

a) 38 47 56

b) $x=9, y=1$ sau $x=149, y=201$

c)

Varianta C/C++

```
#include<iostream.h>
void main()
{
    unsigned_int x, y, aux;
    cout<<"x=";
    cin>>x;
    cout<<"y=";
    cin>>y;
    x=x%10;
    y=y%10;
    if (y<x)
    {
        aux=y;
        y=x;
        x=aux;
    }
    while (x<=y)
    {
        cout<<x*10+y<<" ";
        x=x+1;
        y=y-1;
    }
}
```

d)

se înlocuiește secvența `cat_timp` cu:

```
pentru k=x, y execută
    scrie k*10+y
    y←y-1
sfarsit_pentru
```

SUBIECTUL II

1. b

2. c

3. 9 pentru intensiv

Varianta PASCAL

```
var x, y, aux:word;
begin
    write('x=');
    read(x);
    write('y=');
    read(y);
    x:=x mod 10;
    y:=y mod 10;
    if (y<x) then
    begin
        aux:=y;
        y:=x;
        x:=aux;
    end;
    while (x<=y) do
    begin
        write(x*10+y, ' ');
        x:=x+1;
        y:=y-1;
    end;
end.
```

pentru neintensiv

4		
4	100	
4	100	200

4. 111111 pentru intensiv, respectiv asta pentru neintensiv

5. Exemplul dat în enunțul problemei nu corespunde cu cerința, deoarece numărul 1 nu este

număr par. Așadar primul element al matricei ar trebui să fie 2.

Varianta C/C++

```
#include<iostream.h>
void main()
{
    n,i,j, unsigned int, nr;
    do{
        cout<<"dati n :";
        cin>>n;
    }while(n>20);
    nr=0;
    for(i=1;i<=n;i++)
        for(j=1;j<=n;j++)
        {
            nr=nr+2;
            while(nr%3==0)
                nr=nr+2;
            a[i][j]=nr;
        }
    for(i=1;i<=n;i++)
    {
        for(j=1;j<=n;j++)
            cout<<a[i][j]<<" ";
        cout<<endl;
    }
}
```

Varianta PASCAL

```
var n,i,j,nr:word;
    a:array[1..20,1..20] of word;
begin
    repeat write('dati n :');
           read(n);
    until(n<=20);
    nr:=0;
    for i:=1 to n do
        for j:=1 to n do
            begin
                nr:=nr+2;
                while(nr mod 3=0) do
                    nr:=nr+2;
                a[i,j]:=nr;
            end;
        end;
    for i:=1 to n do
    begin
        for j:=1 to n do
            write(a[i,j], ' ');
        writeln;
    end;
end.
```

SUBIECTUL III

1. a

2. 9, 10 pentru **varianta intensiv**;

pentru neintensiv: pentru a) rezultatul este: 4 pentru b) rezultatul: 9 10

3. a)

Varianta C/C++

Pentru intensiv

```
int max(int x[100], unsigned int n)
{
    if((x[2]-x[1])>0)
        return x[n];
    else
        return x[1];
}
```

Varianta PASCAL

```
type vect=array[1..100] of integer;
function max( x:vect;n:word)
:integer;
begin
    if (x[2]-x[1])>0 then
        max:=x[n]
    else
        max:=x[1];
end;
```

b) Am utilizat proprietățile progresiilor aritmetice, și anume: dacă $a_2 - a_1 > 0$ atunci termenul maxim este a_n , altfel a_1

c) Varianta C/C++

```
#include<fstream.h>
int max(int [], unsigned int);
void main()
{
    int x[100], maxim, maxim1, r;
    unsigned int i, j, ok, n, k;
    ifstream fin("numere.txt");
    fin>>n;
    for(i=1;i<=n;i++)
```

Varianta PASCAL

```
type vect=array[1..100] of integer;
function max( x:vect;n:word)
:integer;
begin
    if (x[2]-x[1])>0 then
        max:=x[n]
    else
        max:=x[1];
end;
```



```

    {
        ok=0;
        fin>>x[1]>>x[2];
        r=x[2]-x[1];
        for(j=3;j<=n;j++)
            fin>>x[j];
        for(j=3;j<=n;j++)
            if(x[j]!=(x[j-1]+r))
            {
                ok=1;
                break;
            }
        if(ok==0)
            break;
    }
    if(i<=n )
        maxim=max(x,n);
        for(k=i+1;k<=n;k++)

        { ok=0;
        fin>>x[1]>>x[2];
        r=x[2]-x[1];
        for(j=3;j<=n;j++)
            fin>>x[j];
        for(j=3;j<=n;j++)
        {
            if(x[j]!=(x[j-1]+r))
            {
                ok=1;
                break;
            }
        }
        if(ok==0)
        {
            maximl=max(x,n);
            if(maxim<maximl)
                maxim=maximl;
        }
    }
    cout<<maxim;
}

```

```

var  maxim,maximl,r:integer;
x:vect;
i,j,ok,n,k:word;
fin:text;
begin
    assign(fin,'numere.txt');
    reset(fin);
    read(fin,n);
    for i:=1 to n do
        begin
            ok:=0;
            read(fin,x[1],x[2]);
            r:=x[2]-x[1];
            for j:=3 to n do
                read(fin,x[j]);
            for j:=3 to n do
                if(x[j]<>(x[j-1]+r)) then
                    begin
                        ok:=1;
                        break;
                    end;
            if(ok=0) then
                break;
        end;
    if(i<=n ) then
        maxim:=max(x,n);
        for k:=i+1 to n do
            begin
                ok:=0;
                read(fin,x[1],x[2]);
                r:=x[2]-x[1];
                for j:=3 to n do
                    read(fin,x[j]);
                for j:=3 to n do
                    begin
                        if(x[j]<>(x[j-1]+r)) then
                            begin
                                ok:=1;
                                break;
                            end;
                    end;
            if(ok=0) then
                begin
                    maximl:=max(x,n);
                    if(maxim<maximl) then
                        maxim:=maximl;
                end;
        end;
    end;
    write(maxim);
end.

```

3) Pentru neintensiv:

În exemplu se afișează și 709, lucru care este greșit

a) int cifre_impere(unsigned long int n)

b) Varianta C/C++

```

#include<istream.h>
int cifre_impere(unsigned long int
);

```

Varianta PASCAL

```

function cifre_impere(
n:longint):integer;
var ok:integer;

```

```

void main()
{
    ifstream fin("numere.txt");
    unsigned int n,i;
    unsigned long int nr;
    fin>>n;
    for(i=1;i<=n;i++)
    {
        fin>>nr;
        if(nr>100 &&
cifre_impere(nr)==1)
            cout<<nr<<" ";
    }
}
int cifre_impere(unsigned long int
n)
{
    int ok=1;

    while(n)
        if(n%10%2==0)
        {
            ok=0;
            break;
        }
        n=n/10;
    }
    return ok;
}

```

4. Neintensiv:

În enun nu se specific valoarea maxim pe care o poate avea **n**, fapt care poate duce la confuzionarea elevilor. ă

Varianta C/C++

```

#include<iostream.h>
void main()
{
    unsigned int
n,i,v[100],lmax,l,k;
    cout<<"dati n:";
    cin>>n;
    for(i=1;i<=n;i++)
    do{
        cout<<"v["<<i<<"]="";
        cin>>v[i];

    }while(v[i]>999);

    for(i=1;i<=n;i++)
        if(v[i]%10%2==0)
            break;
    l=1;
    lmax=1;
    for(k=i+1;k<=n;k++)
        if(v[k]%10%2==0)
            l++;
    else
    {
        if(l>lmax)
            lmax=l;
    }
}

```

```

begin
    ok:=1;
    while(n<>0) do
    begin
        if(n mod 10 mod 2=0) then
        begin
            ok:=0;
            break;
        end;
        n:=n div 10;
    end;
    cifre_impere:=ok;
end;
var fin:text;
    n,i:word;
    nr:longint;
begin
    assign(fin,'numere.txt');

    read(fin,n);
    for i:=1 to n do
    begin
        read(fin,nr);
        if(nr>100) and
(cifre_impere(nr)=1) then
            write(nr,' ');
        end;
    end.
end.

```

Varianta PASCAL

```

var    n,i,lmax,l,k:word;
        v:array[1..100] of word;
begin
    write('dati n:');
    read(n);
    for i:=1 to n do
    repeat
        write('v['<i>','<i>']='');
        read(v[i]);

    until (v[i]<=999);

    for i:=1 to n do
        if (v[i] mod 10 mod 2=0) then
            break;
    l:=1;
    lmax:=1;
    for k:=i+1 to n do
        if(v[k] mod 10 mod 2=0) then
            l:=l+1
        else
        begin
            if(l>lmax) then
                lmax:=l;
            l:=0;
        end;
    end;
end;

```

```

        l=0;
    }
    if(l>lmax)
        cout<<l;
    else
        cout<<lmax<<endl;
}

```

```

    if(l>lmax) then
        write(l)
    else
        writeln(lmax);
end.

```

Varianta <35>

SUBIECTUL I

1. c

2.

a) 4 programul calculează suma puterilor factorilor primi din descompunerea lui x

b) $x=64$

c)

Varianta C/C++

```

#include<iostream.h>
void main()
{
    unsigned int x,s,f,p;
    cout<<"x=";
    cin>>x;
    s=0;
    f=2;
    while(x>1)
    {
        p=0;
        while(x%f==0)
        {
            x=x/f;
            p++;
        }
        if(p!=0)
            s=s+p;
        f=f+1;
    }
    cout<<s;
}

```

Varianta PASCAL

```

var x,s,f,p:word;
begin
    write('x=');
    read(x);
    s:=0;
    f:=2;
    while(x>1) do
    begin
        p:=0;
        while(x mod f =0) do
        begin
            x:=x div f;
            p:=p+1;
        end;
        if(p<>0) then
            s:=s+p;
        f:=f+1;
    end;
    write(s);
end.

```

d) numerele prime din intervalul dat, adică: 7,11,13,17,19,23

SUBIECTUL II

1. d

2. c

3. **amaara** pentru intensiv; pentru neintensiv: **6iar** pe a doua linie **amat**

4. 5 componente conexe, deci vom adăuga 4 muchii

5.

Varianta C/C++

```

#include<iostream.h>
void main()
{
    unsigned int n,i,j;
    int a[50][50],min;
    do{

```

Varianta PASCAL

```

var n,i,j:word;
    a:array[1..50,1..50] of
integer;min:integer;
begin
    repeat
        write('n=');

```

```

        cout<<"n=";
        cin>>n;
    }while(n>50);
    for(i=1;i<=n;i++)
        for(j=1;j<=n;j++)
        {
            cout<<"a["<<i<<"] ["<<j<<"]=";
            cin>>a[i][j];
        }
    for(j=1;j<=n;j++)
    {
        min=a[1][j];
        for(i=2;i<=n;i++)
            if(min>a[i][j])
                min=a[i][j];
        cout<<min<<" ";
    }
}

```

```

        read(n);
    until(n<=50);
    for i:=1 to n do
        for j:=1 to n do
            begin
                write('a[' ,i, ']' [,j, ']=');
                read(a[i,j]);
            end;
        for j:=1 to n do
            begin
                min:=a[1,j];
                for i:=2 to n do
                    if(min>a[i,j]) then
                        min:=a[i,j];
                write(min, ' ');
            end;
        end.

```

SUBIECTUL III

1. c

2. pentru varianta **intensiv**: 42-1-3

pentru **neintensiv** 3, iar ca exemplu orice șir de numere descrescător

3.

Varianta C/C++

Pentru intensiv

```

#include<fstream.h>
void main()
{
    ifstream fin("numere.in");
    unsigned int n,i;
    unsigned long int nr,x,y;
    fin>>n;
    for(i=1;i<=n;i++)
    {
        fin>>nr;
        x=nr;
        if(x<=9)
            cout<<nr<<" ";
        else
        {
            y=nr%10;
            x=x/10;
            while(x>9)
                x=x/10;
            if(x==y)
                cout<<nr<<" ";
        }
    }
    fin.close();
}

```

3. Pentru **neintensiv**:

Varianta C/C++

a) vom adăuga un parametru funcției, ce va asigura recursia.

```

unsigned int sum(unsigned int
x,unsigned int i)

```

Varianta PASCAL

```

var fin:text;
    n,i:word;
    nr,x,y:longint;
begin
    assign(fin,'numere.in');
    reset(fin);
    read(fin,n);
    for i:=1 to n do
        begin
            read(fin,nr);
            x:=nr;
            if(x<=9) then
                write(nr, ' ');
            else
                begin
                    y:=nr mod 10;
                    x:=x div 10;
                    while(x>9) do
                        x:=x div 10;
                    if(x=y) then
                        write(nr, ' ');
                end;
        end;
    close(fin);
end.

```

Varianta PASCAL

```

function sum(x,i:word) :word;
begin

```

```

{
    if(i>1)
        if(x%i==0)
            return
i+sum(x,i-1);
        else
            return sum(x,i-
1);
    else
        return 0;
}

```

```

b) #include<iostream.h>
unsigned int sum(unsigned
int,unsigned int);
void main()
{
    unsigned int i,n,j,nr,v[100],aux;
    do{
        cout<<"n=";
        cin>>n;
    }while(n==0|| n>=100);
    for(i=1;i<=n;i++)
    {
        do{
            cout<<"nr=";
            cin>>nr;
        }while(nr>9999);
        v[i]=sum(nr,nr/2);
    }
    for(i=1;i<=n;i++)
        for(j=i+1;j<=n;j++)
            if(v[i]>v[j])
            {
                aux=v[i];
                v[i]=v[j];
                v[j]=aux;
            }
    for(i=1;i<=n;i++)
        cout<<v[i]<<" ";
}

```

4. Pentru intensiv

a) Varianta C/C++

```

unsigned int sum(unsigned int x)
{
    unsigned int i,s=0;
    for(i=2;i<=x/2;i++)
        if(x%i==0)
            s=s+i;
    return s;
}

```

```

b) #include<iostream.h>
unsigned int sum(unsigned int);
void main()
{
    unsigned int i,n,j,nr,v[100],aux;
    do{

```

```

        if(i>1) then
            if(x mod i=0) then
                sum:=i+sum(x,i-
1)
            else
                sum:=sum(x,i-1)
        else
            sum:=0;
    end;

```

Se completeazăcu:

```

var i,n,j,nr,aux:word;
    v:array[1..100] of word;
begin
    repeat
        write('n=');
        read(n);
    until(n>0) and (n<=100);
    for i:=1 to n do
        begin
            repeat
                write('nr=');
                read(nr);
            until(nr<=9999);
            v[i]:=sum(nr,nr div 2);
        end;
        for i:=1 to n do
            for j:=i+1 to n do
                if(v[i]>v[j]) then
                    begin
                        aux:=v[i];
                        v[i]:=v[j];
                        v[j]:=aux;
                    end;
        for i:=1 to n do
            write(v[i],' ');
    end.

```

Varianta PASCAL

```

function sum(x:word):word;
var i,s:word;
begin
    s:=0;
    for i:=2 to x div 2 do
        if(x mod i=0) then
            s:=s+i;
    sum:=s;
end;

```

b) se completează cu:

```

var i,n,j,nr,aux:word;
    v:array[1..100] of word;
begin
    repeat
        write('n=');

```

```

        cout<<"n=";
        cin>>n;
    }while(n==0|| n>=100);
    for(i=1;i<=n;i++)
    {
        do{
            cout<<"nr=";
            cin>>nr;
        }while(nr>9999);
        v[i]=sum(nr);
    }
    for(i=1;i<=n;i++)
        for(j=i+1;j<=n;j++)
            if(v[i]>v[j])
            {
                aux=v[i];
                v[i]=v[j];
                v[j]=aux;
            }
    for(i=1;i<=n;i++)
        cout<<v[i]<<" ";
}

```

4. neintensiv

Varianta C/C++

```

#include<fstream.h>
void main()
{
    ifstream fin("numere.in");
    unsigned int n,i,j,v[11],s,k;
    unsigned long int
nr,x;
    fin>>n;
    for(i=1;i<=n;i++)
    {
        fin>>nr;
        x=nr;
        for(j=1;j<=10;j++)
            v[j]=0;
        while(x)
        {
            v[x%10]++;
            x=x/10;
        }
        s=0;
        k=0;
        for(j=1;j<=10;j++)
            if(v[j]==1)
                k++;
            if(k==3)
                cout<<nr<<" ";
        }
    fin.close();
}

```

```

        read(n);
    until(n>0) and (n<100);
    for i:=1 to n do
    begin
        repeat
            write('nr=');
            read(nr);
            until (nr<=9999);
            v[i]:=sum(nr);
        end;
        for i:=1 to n do
            for j:=i+1 to n do
                if(v[i]>v[j]) then
                    begin
                        aux:=v[i];
                        v[i]:=v[j];
                        v[j]:=aux;
                    end;
            end;
        for i:=1 to n do
            write(v[i], ' ');
        end.

```

Varianta PASCAL

```

var fin:text;
    n,i,j,s,k:word;
    v:array[1..11] of word;
    nr,x:longint;
begin
    assign(fin,'numere.in');
    reset(fin);
    read(fin,n);
    for i:=1 to n do
    begin
        read(fin,nr);
        x:=nr;
        for j:=1 to 10 do
            v[j]:=0;
        while(x<>0) do
            begin
                v[x mod 10]:=v[x mod
10]+1;
                x:=x div 10;
            end;
            s:=0;
            k:=0;
            for j:=1 to 10
do
                if(v[j]=1)
then
                    k:= k+1;
                    if(k=3) then
                        write(nr,' ');
                    end;
            close(fin);
        end.

```

SUBIECTUL I

- 1. b
- 2. a) 249

b) ~~4950~~₀

```
citeste v
[   executa
    a=v%10
    b=[v/10]%10
    s=s+a*10+b
    daca v>0 atunci citeste v
]pana cand v=0
scrie s
```

d)

Varianta C/C++

```
#include<iostream.h>
void main()
{int s,a,b,v;
s=0; cin>>v;
while(v!=0)
    {a=v%10;      b=v/10%10;
    s=s+a*10+b;   cin>>v; }
cout<<s;}
```

Varianta PASCAL

```
var s,v,a,b :integer;
begin
    s:=0;
    write('v=');read(v);
    while v<>0 do
        begin a:= v mod 10;
            b:=(v div 10) mod 10;
            s:=s+a*10+b;
            write('v=');read(v); end;
    write('s=',s);end.
```

SUBIECTUL II

- 1. b pentru varianta intensiv, respectiv a pentru neintensiv
- 2. c pentru varianta intensiv, respectiv b pentru neintensiv
- 3. 3 pentru varianta intensiv([2,7],[1,7],[1,6]), respectiv 3 pentru neintensiv([1,2],[1,3],[2,3])
- 4. 152(18*19/2 – 19=152)
- 5.

Varianta C/C++

```
#include<iostream.h>
void main()
{int a[31][31],n,m,i,j,min;
cout<<"n="; cin>>n; cout<<"m=";

cin>>m;
for(i=1;i<=n;i++)
    for(j=1;j<=m;j++)
    {cout<<"a["<<i<<"]["<<j<<"]=";cin>>a[i][j];}
for(j=1;j<=m;j++) {min=a[1][j];
    for(i=2;i<=n;i++)
        if(a[i][j]<min)
            min=a[i][j];
    cout<<min<<" ";
}}
```

Varianta PASCAL

```
var n,m,i,j,min :integer;
    mat:array [1..10,1..10] of
integer;
begin
write('n=');read(n);write('m=');re
ad(m);
for i:=1 to n do
    for j:=1 to m do begin
        write ('mat['<i>','<j>','<j>']='');
        read(mat[i,j]); end;
for j:=1 to m do begin
    min:=mat[1,j];
    for i:=2 to n do
        if mat[i,j]<min then
            min:=mat[i,j];
    write (min,' '); end;
end.
```

SUBIECTUL III

- 1. b pentru varianta intensiv, respectiv a pentru neintensiv

2. $2+3+7; 2+4+6; 2+10$

3. Pentru intensiv:

a) Varianta C/C++

```
int cifra(int a)
{while(a)
{if(a%2==0) return a%10;
a=a/10;}
return -1;}
```

```
b) #include<iostream.h>
#include<fstream.h>
int n, v[10],i,a,c;
fstream f("bac.in", ios::in);
int cifra(int a)
{ if(a==0) return 0;
while(a)
{if(a%2==0) return a%10;
a=a/10;}
return -1;}
void main()
{f>>n;
for(i=1;i<=n;i++) {f>>a;
c=cifra(a);
if(c>=0) v[c]++; }
for(i=9;i>=0;i--)
while (v[i]) {cout<<i; v[i]--
;}}
```

Varianta PASCAL

```
function cif (x: integer):integer;
var c:integer;
begin c:=-1;
if x=0 then c:=0
else while ((x>0) and (c=-1)) do
begin
if x mod 2= 0 then c:=x mod 10;
x:=x div 10; end;
cif:=c;end;
var n,i,val,c :integer; f: text;
v: array [0..9] of integer;
function cif (x: integer):integer;
var c:integer;
begin
if x=0 then c:=0 else begin
c:=-1;
while ((x>0) and (c=-1)) do begin
if x mod 2= 0 then c:=x mod 10;
x:=x div 10; end;
end;
cif:=c; end;
begin
assign(f,'bac.in');reset(f);read(f,n);
for i:=0 to 9 do v[i]:=0;
for i:=1 to n do begin
read(f,val); c:=cif(val);
for i:=1 to c do inc(v[c]);end;
while v[i]>0 do begin
write (i); dec(v[i]); end;
end.
```

c) Cele **n** numere au fost prelucrate fără fi memorate, rezultatele (ultima cifră apară) au fost memorate într-un vector de cifre (0..9) apoi s-a parcurs vectorul 9->0 și s-au afișat cifrele. Nu s-au folosit vectori de dimensiuni mari și nu s-a facut ordonarea cifrelor rezultate. Complexitate de ordin $O(n)$.

3. Pentru neintensiv:

Varianta C/C++

```
#include<iostream.h>
void main()
{int i,n;
cout<<"n="; cin>>n;
for(i=1;i<=n;i++) cout<<5*i<<"
"; }
```

Varianta PASCAL

```
var n,i :integer;
begin
write('n=');read(n);
for i:=1 to n do
write (5*i,' ');
end.
```

4. Pentru neintensiv: O variantă de rezolvare pentru subprogramul de la punctul **a** și programul de la punctul **bg** așiți la varianta intensiv subiectul **III.3** punctele **a** și **b**.

Varianta <37>

SUBIECTUL I

1. a) 122322

b) 874 și 5 Observație: **k** diferit de cifrele numărului **n**


```

c) citeste n,k
   nr=0; p=1
   executa
   |   c=n%10
   |   nr=nr+c*p
   |   p=p*10
   |   -daca c=k atunci
   |   |   nr=nr+c*p
   |   |   p=p*10
   |   -
   |   n=n/10
   -pana cand n=0
n=nr
scrie n
    
```

d)

Varianta C/C++

```

#include<iostream.h>
void main()
{int n,p,k,c,nr;
cin>>n>>k; nr=0; p=1;
while(n!=0)
    {c=n%10; nr=nr+c*p;    p=p*10;
    if(c==k){nr=nr+c*p; p=p*10;}
    n=n/10;  }
n=nr; cout<<n;}
    
```

Varianta PASCAL

```

var n,nr,p :longint;
c,k:integer;
begin
    write('n='); read(n); write('k=');
    read(k);
    nr:=0; p:=1;
    while n<>0 do begin
        c:=n mod 10; nr:=nr+c*p;
        p:=p*10;
        if c=k then begin
            nr:=nr+c*p; p:=p*10; end;
            n:=n div 10; end;
    n:=nr; writeln ('n=',n); end.
    
```

SUBIECTUL II

1. d pentru varianta intensiv, respectiv b pentru neintensiv

2. b pentru varianta intensiv, respectiv a pentru neintensiv

3. 01011

10100

00010

00001

10000

4. 2 și 3

5. Varianta C/C++

```

#include<iostream.h>
#include<string.h>
void main()
{char sir[256];
int i,x;
cin.get(sir,256);
x=strlen(sir);
if(sir[0]!=' ') sir[0]=sir[0]-32;
if(sir[x-1]!=' ')sir[x-1]=sir[x-1]-32;
for(i=1;i<=x-2;i++)
if(sir[i-1]==' '||sir[i+1]==' ')
    sir[i]=sir[i]-32;
cout<<sir;}
    
```

Varianta PASCAL

```

var sir: string; i,lung :integer;
begin
write ('Sirul este: ');read(sir);
lung:=length(sir);
if sir[1]<>' ' then
    sir[1]:=chr(ord(sir[1])-32);
if sir[lung]<>' ' then
    sir[lung]:=chr(ord(sir[lung])-32);
for i:=2 to lung-1 do begin
    if sir[i-1]=' ' then
        sir[i]:=chr(ord(sir[i])-32);
    if (sir[i+1]=' ')and(sir[i-1]<>' ') then
        sir[i]:=chr(ord(sir[i])-32);
end;
write(sir);end.
    
```

SUBIECTUL III

1. c pentru varianta intensiv, respectiv b pentru neintensiv

2. -11

3.

Varianta C/C++

```
#include<iostream.h>
void main()
{int n,v[100],k,i,j,aux;
cout<<"n="; cin>>n;
cout<<"k="; cin>>k;
for(i=1;i<=n;i++)
{cout<<"v["<<i<<"]="; cin>>v[i];}
for(j=1;j<=k;j++)
{aux=v[1];
for(i=1;i<=n;i++) v[i]=v[i+1];
v[n]=aux;}
for(i=1;i<=n;i++) cout<<v[i]<<" ";
}
```

4 a) int div(int x)

b) Pentru intensiv:

Varianta C/C++

```
#include<fstream.h>
#include<iostream.h>
fstream f("bac.in", ios::in);
int div(int x)
{int nr=0,i=2;
while(x>1)
{if(x%i==0){nr++;
while(x %i== 0) x/=i; }
i++; }
return nr;}
void main()
{int n,p=-1,u,v[100],i;
f>>n;
for(i=1;i<=n;i++)
f>>v[i];
for(i=1;i<=n;i++)
if(div(v[i])%2==0)
{if(p==-1) p=v[i];u=v[i];}
cout<<p<<" "<<u;}
}
```

Pentru neintensiv:

Varianta C/C++

```
#include<iostream.h>
#include<fstream.h>
fstream f("bac.in", ios::in);
int div(int x)
{int nr=0,i=2;
while(x>1)
{if(x%i==0){nr++;
while(x%i== 0) x/=i; }
i++; }
return nr;}
void main()
}
```

Varianta PASCAL

```
var n,k,aux,i,j :integer;
v : array [1..100] of integer;
begin
write ('n=');read(n);
for i:=1 to n do begin
write('\v[' ,i,']=');read (v[i]); end;
write ('k=');read(k);
for j:=1 to k do begin
aux:=v[1];
for i:=1 to n-1 do v[i]:=v[i+1];
v[n]:=aux; end;
for i:=1 to n do
write(v[i], ' ');
end.
```

function div(x: integer): byte;

Varianta PASCAL

```
var n,i,val,p,u :integer; f: text;
v : array [0..9] of integer;
function div1 (x: integer):integer;
var nr,i:integer;
begin
nr:=0;i:=2;
while x>1 do begin
if x mod i =0 then begin inc(nr);
while x mod i=0 do x:=x div i; end;
i:=i+1; end;
div1:=nr; end;
begin
assign(f, 'bac.in');reset(f);read(f,n);
p:=-1; u:=-1;
for i:=1 to n do begin
read(f,val);
if div1(val) mod 2 =0 then begin
if p=-1 then p:=val;
u:=val; end;
end;
write (p, ' ',u); end.
```

Varianta PASCAL

```
var n,i,val :integer; f: text;
v : array [0..9] of integer;
function div1 (x: integer):integer;
var nr,i:integer;
begin nr:=0;i:=2;
while x>1 do begin
if x mod i =0 then begin
inc(nr);
while x mod i= 0 do x:=x div i;
end;
i:=i+1; end;
```

```
{int n, v[100],i;
f>>n;
for(i=1;i<=n;i++) f>>v[i];
for(i=1;i<=n;i++)
cout<<div(v[i])<<" ";}
```

```
div1:=nr; end;
begin
assign(f,'bac.in');reset(f);read(f,n);
for i:=1 to n do begin
read(f,val);write(div1(val),' ');
end; end.
```

Varianta <38>

SUBIECTUL I

1. d

2. a) 4

b) 12381și 1

c) citește n, k
 pentru i=k, 0, -1 executa n=[n/10]
 z=n%10
 scrie z

d)

Varianta C/C++

```
#include<iostream.h>
void main()
{ int i,k,z,n;
cin>>n>>k; i=k;
while(i>0) {n=n/10; i- -;}
z=n%10; cout<<z;}
```

Varianta PASCAL

```
var n,k,z,i:integer;
begin
write('n='); read(n); write('k=');
read(k);
i:=k;
while i>0 do begin
n:=n div 10; i:=i-1; end;
z:=n mod 10; write(z); end.
```

SUBIECTUL II

1. c pentru **varianta intensiv**, respectiv a pentru **neintensiv**

2. d pentru **varianta intensiv**, respectiv c pentru **neintensiv**

3. 2 ([1,5],[5,6])

4. 4

5. Varianta C/C++

```
#include<iostream.h>
void main()
{int n, a[30][30],i,j,min,p=1;
cout<<"n="; cin>>n;
for(i=1;i<=n;i++)
for(j=1;j<=n;j++)
{cout<<"a["<<i<<"["<<j<<"]=";
cin>>a[i][j];}
for(j=1;j<=n;j++)
{min=a[1][j];
for(i=2;i<=n;i++)
if(a[i][j]<min)
min=a[i][j];
if (min == a[n+1-j][j])
p=p*min;
}
cout<<p;
}
```

Varianta PASCAL

```
type mat=array[1..10,1..10] of
integer;
var n,i,j,min :integer; p:
longint; a:mat;
begin
write('n='); read(n);
for i:=1 to n do
for j:=1 to n do begin
write('a['<i','<j,']=');
read (a[i,j]); end;
p:=1;
for j:=1 to n do begin
min:=a[1,i];
for i:=2 to n do if
a[i,j]<min then
min:=a[i,j];
if min=a[n+1-j,j] then
p:=p*min; end;
write('p=',p); end.
```

SUBIECTUL III

1. d pentru **varianta intensiv**, respectiv c pentru **neintensiv**

2. 19

3.

Variantă C/C++

```
void del (longint &x, int y)
{ longint p=1, x0=0;
  while(x)
  { if(x % 10 <= y) { x0=x0+p*(x %
10);
    p=p*10;    }
    x=x/10;  }
  if(x0==0) x=-1; else x=x0;}
```

4. a)

Variantă C/C++

```
void inter(int& x, int &y)
```

b)

Variantă C/C++

```
#include<fstream.h>
fstream f("bac.in", ios::in);
void inter(int& x, int &y)
{int aux;
aux=x;
x=y;
y=aux;
}
void main()
{int n, v[1000],i,j;
f>>n;
for(i=1;i<=n;i++)
f>>v[i];
for(i=1;i<n;i++)
for(j=i+1;j<=n;j++)
if(v[i]>v[j])
inter(v[i],v[j]);
for(i=1;i<=n;i++)
cout<<v[i]<<" ";
```

Variantă PASCAL

```
procedure del (var x: longint;
y:integer);
var p,x0:longint;
begin x0:=0; p:=1;
while x>0 do begin
if (x mod 10)<=y then begin
x0:=x0+p*(x mod 10); p:=p*10;
end;
x:=x div 10; end;
if x0=0 then x:=-1 else x:=x0; end;
```

Variantă PASCAL

```
procedure inter(var x:integer;var
y:integer);
```

Variantă PASCAL

```
var n,i,j,aux:integer;
v:array[1..100] of integer;
f: text;
procedure inter(var x:integer;var
y:integer);
var aux:integer;
begin aux:=x; x:=y; y:=aux;end;
begin
assign(f,'bac.in'); reset(f);
read(f,n);
for i:=1 to n do read(f,v[i]);
for i:=1 to n-1 do
for j:=i+1 to n do
if v[i]>v[j] then
inter(v[i],v[j]);
for i:=1 to n do write(v[i],' ');
end.
```

Variantă <39>

SUBIECTUL I

1. b

2. a) 23949

b) 999

c)

```

citeste n
nr=0 p=1
executa
  c=n%10
  daca c<9 atunci
    c=c+1
  nr=nr+c*p
  p=p*10
  n=n/10
pana cand n=0
n=nr
scrie n
    
```

d)

Varianta C/C++

```

#include<iostream.h>
void main()
{int nr,n,p,c;
cin>>n; nr=0; p=1;
while(n!=0) {c=n%10;
              if(c<9) c=c+1;
              nr=nr+c*p;
              p=p*10; n=n/10;}
n=nr; cout<<n; }
    
```

Varianta PASCAL

```

var n,nr,p,c:integer;
begin
write('n='); read(n); nr:=0;
p:=1;
while n<>0 do begin
c:=n mod 10;
if c<9 then c:=c+1;
nr:=nr+c*p; n:=n div 10;
p:=p*10;
end;
n:=nr; write(n);end.
    
```

SUBIECTUL II

1. b pentru varianta intensiv, respectiv c pentru neintensiv
2. c pentru varianta intensiv, respectiv a pentru neintensiv
3. 144 pentru varianta intensiv, respectiv 3 pentru neintensiv(1,5 – 2 – 3,4)
4. 234
345
456

5. Varianta C/C++

```

#include<iostream.h>
#include<string.h>
char a[100];
void main()
{char aux; int i,k1,k2,n,j;
cin.get(a,99); n=strlen(a);
a[n]='*';
for(i=0;i<n;i++)
if((a[i]=='a' || a[i]=='e' || a[i]=='o' ||
a[i]=='i' || a[i]=='u') && (a[i-1]=='
' || i-1== -1))
{k1=i; while(a[i+1]!='
' &&
a[i+1]!='*')
{i++;k2=i;}
for(j=0;j<=(k2-k1)/2;j++)
{aux=a[k1+j];
a[k1+j]=a[k2-j];
    
```

Varianta PASCAL

```

var sir: string; aux: char;
i,j,k, lung, lung2 :integer;
begin
write ('Sirul este: '); read(sir);
lung:=length(sir);
for i:=1 to lung do begin
if (((i=1) or ((i>1) and (sir[i-1]='
')))
and (sir[i] in ['a','e','i','o','u']))
then begin
k:=i;
while (sir[k]<>'
') and (k<lung) do
inc(k);
lung2:=k-i; k:=k-1;
for j:=i to i+lung2 div 2-1 do
begin
    
```

```

        a[k2-j]=aux;}
    } a[n]=NULL;
    cout<<a; }

```

```

        aux:=sir[j];
    sir[j]:=sir[k-(j-i)];
        sir[k-(j-i)]:=aux; end;
    end; end;
    write(sir); end.

```

SUBIECTUL III

1. b pentru **varianta intensiv**, respectiv d pentru **neintensiv**

2. 9

3.

Variantă C/C++

```

#include<fstream.h>
fstream f("bac.in", ios::in);
void main()
{int n, v[1000],i,x,ok;
f>>n;
for(i=1;i<=n;i++)
f>>v[i];
ok=1;x=-1;
for(i=1;i<=n;i++)
{if(v[i]%2==0)
    if(v[i]<x) ok=0;
    else x=v[i];}
if(ok) cout<<"DA"; else cout<<"NU";
}

```

4. a) int pr(long a)

b) Variantă C/C++

```

#include<iostream.h>
int pr(long a)
{int i;
for(i=2;i<a/2;i++)
    if(a%i==0) return 0;
return 1;}
void main()
{long n; cin>>n;
while(n) {n=n/10;
    if(pr(n)) if(n>1) cout<<n<<" ";
}}

```

Variantă PASCAL

```

var n,i,cresc :integer;
    v: array [1..1000] of integer;
    f: text;
Begin
    assign(f,'bac.in'); reset(f);
    read(f,n);
    for i:=1 to n do
        read(f,v[i]);
    cresc:=1;
    for i:=1 to n-1 do
        if v[i]>v[i+1] then cresc:=0;
    if cresc=1 then write('DA')
    else write('NU');
End.

```

Variantă PASCAL

```

var n :longint;
function pr (a: longint):boolean;
    var i:integer; prim:boolean;
begin
    prim:=true;
    for i:=2 to a div 2 do
        if a mod i = 0 then prim:=false;
    pr:=prim; end;
begin write('n='); read(n);
    while n>99 do begin
        n:=n div 10;
        if pr(n)=true then write(n,'
');
    end; end.

```

Variantă <40>

SUBIECTUL I

1. c

2. a) 2 și 7

b) 3 sau 9 sau 16 (un număr cu un singur factor prim $n=f^k$)

c) citește x

d=2 y=0 z=0

```

┌   executa
├   p=0
└   ┌   executa

```

```

┌   ┌   R=P*x/d]
├   └   pana cand x%d >0
└   ┌   └   ┌   executa

```

```

    |
    |         daca y=0  atunci y=d
    |         z=d
    |
    |         d=d+1
    |
    |         pana cand x=1
    |         scrie y scrie z

```

d)

Varianta C/C++

```

#include<iostream.h>
void main()
{int x,d,y,z,p;
cin>>x; d=2; y=0; z=0;
while(x!=1) {p=0;
while(x%d==0)
    {p=p+1;      x=x/d;      }
if(p!=0) {if(y=0) y=d;
z=d;      }
d=d+1;}
cout<<y<<" "<<z;}

```

Varianta PASCAL

```

var d,p,x,y,z :integer;
Begin
write('x=');read(x);
d:=2;y:=0;z:=0;
while x<>1 do begin
p:=0;
while x mod d =0 do begin
p:=p+1; x:=x div d;
end;
if p<>0 then begin
if y=0 then y:=d;
z:=d; end;
d:=d+1; end;
writeln('y=',y); writeln('z=',z);
End.

```

SUBIECTUL II

- 1. d pentru **varianta intensiv**, respectiv **a pentru neintensiv**
- 2. b pentru **varianta intensiv**, respectiv **d pentru neintensiv**
- 3. 3 componente (1,5 - 2 - 3,4,7,8)
- 4.111

122
123

5. Varianta C/C++

```

#include<iostream.h>
#include<string.h>
char sir[256],sir2[20],*p;
int lcuvant ;

void main()
{ cout<<"Sirul este:";
cin.get(sir,255);
p=strchr(sir,'*');
strcpy(sir2,sir);
sir2[p-sir]='\0';
lcuvant=strlen(sir2);
p=strstr(sir,sir2);
while (p)
{ strcpy(p,p+lcuvant);
p=strstr(sir,sir2);      }
cout<<sir; }

```

Varianta PASCAL

```

var sir,sir2: string;
i,j,lung,lung2 :integer;
begin
write ('Sirul este: '); read(sir);
lung:=length(sir); i:=1;
while sir[i+1]<>'*' do i:=i+1;
lung2:=i; sir2 :=copy(sir,1,i);
delete (sir,1,lung2); lung:=lung-
lung2;
i:=1;
while i<=lung do begin
if sir[i]='*' then begin
j:=i+1;
while sir[j]=sir2[j-i] do
j:=j+1;
j:=j-1;
if (j-i=lung2) and
(sir[i+lung2+1]='*') then begin

delete(sir,i+1,lung2);lung:=lung-
lung2;
end;
end;
i:=i+1; end;
write(sir);end.

```

SUBIECTUL III

1. c pentru varianta intensiv, respectiv a pentru neintensiv

2. 12 pentru varianta intensiv respectiv 5 pentru neintensiv

3.

Varianta C/C++

```
#include<fstream.h>
fstream f("bac.in", ios::in);
int n, v[1000], i, x, ok;
void sortare()
{
    int t=0;
    while(!t)
    {
        t=1;
        for(int i=1;i<n;i++)
            if(v[i]>v[i+1])
            {
                int aux=v[i];
                v[i]=v[i+1];
                v[i+1]=aux;
                t=0;
            }
    }
}
void main()
{
    f>>n;
    for(i=1;i<=n;i++) f>>v[i];
    ok=1;x=-1;
    for(i=1;i<=n;i++)
        {if(v[i]%2==0)    if(v[i]<x) ok=0;
                                     else x=v[i];}
    x=10000;
    for(i=1;i<=n;i++)
        {if(v[i]%2!=0) if(v[i]>x) ok=0;
                                     else x=v[i];}
    if(ok) cout<<"DA"; else
    {
        sortare();
        for(i=1;i<=n;i++)
            cout<<v[i]<<" ";
    }
}
```

4. a)

```
int pr(int x)      int div(int y)
```

b) Varianta C/C++

```
#include<iostream.h>
int pr(int x)
{int i;
for(i=2;i<=x/2;i++) if(x%i==0) return
0;
return 1;}
int div(int y)
```

Varianta PASCAL

```
var n,i,uv :integer;
    v: array [1..1000] of integer;
    f: text; ok:boolean;
procedure sortare;
var t,aux,i:integer;
begin
    t:=0;
    while t==0 do
        begin
            t:=1;
            for i:=1 to n-1 do
                if v[i]>v[i+1] then
                    begin
                        aux:=v[i];
                        v[i]:=v[i+1];
                        v[i+1]:=aux;
                        t:=0;
                    end;
            end;
        end;
end;
begin
    assign(f,'bac.in'); reset(f); read(f,n);
    for i:=1 to n do read(f,v[i]);
    ok:=true; uv:=0;
    for i:=1 to n do
        if (v[i] mod 2=0) then
            if v[i]>uv then uv:=v[i]
            else ok:=false;
    uv:=10000;
    for i:=1 to n do
        if (v[i] mod 2=1) then
            if v[i]<uv then uv:=v[i]
            else ok:=false;
    if ok=true then write('DA')
    else
        begin
            sortare;
            for i:=1 to n do
                write(v[i],' ');
        end;
end.
```

end.

```
function pr(x:integer) :byte;
function div1(y:integer) :integer;
```

Varianta PASCAL

```
var n,i :integer;
function pr(x:integer) :byte;
var i,p: integer;
begin p:=1;
for i:=2 to x div 2 do
    if x mod i = 0 then p:=0;
pr:=p; end;
```



```
cout<<a[i][j]<<" ";
cout<<endl;}}
```

```
end; end.
```

SUBIECTUL III

1. a

2. 7 si 5

3. Varianta C/C++

```
int subpr(int a[100],int n)
{int i,j;
for(i=1;i<n;i++)
for(j=i+1;j<=n;j++)
if(a[i]==[aj]) return 0;
for(i=1;i<n;i++)
if(a[i]-a[i+1]==1||a[i]-a[i+1]=-1)
return 0;
return 1;}
```

Varianta PASCAL

```
type vect= array [1..100] of integer;
function
ver(v:vect;n:integer):integer;
var i,j:integer; ok :byte;
begin ok:=1;
for i:=1 to n-1 do
for j:=i+1 to n do
if v[i]=v[j] then ok:=0;
for i:=1 to n-1 do
if abs(v[i]-v[i+1])=1 then
ok:=0;
ver:=ok;
end;
```

4. Pentru intensiv

a) Varianta C/C++

```
#include<fstream.h>
fstream f("numere.txt", ios::in);
void main()
{long int n,nr; int x, max, i;
max=-1; f>>n;
for(i=1;i<=n;i++)
{f>>x;
if(x>max) {max=x; nr++;}
else if(x==max) nr++;}
cout<<max<<" "<<nr;}
```

Varianta PASCAL

```
var n,i :longint; val: byte;
f:text;
v :array [0..9] of longint;
begin
assign(f,'numere.txt'); reset(f);
read (f,n);
for i:=1 to n do begin
end;read(f,val); inc(v[val]);
for i:= 9 downto 0 do
if v[i]>0 then begin write (i,'
',v[i]); break;
end; close(f);end.
```

b) Intensiv C/C++:S-a citit fiecare din cele n cifre fără le memora, s-a comparat cu **max** și numărât de câte ori se găsesc valorile.

In varianta **PASCAL**num ărul de apariții pentru fiecare cifrăa fost stocat într-un vector **v[0..9]**,prelucrarea s-a facut odat ăcu parcugerea (complexitate $O(n)$).

4. Pentru neintensiv

Varianta C/C++

```
#include<fstream.h>
fstream f("numere.txt", ios::in);
void main()
{int n,x, max,i,min;
max=-1;min=10;
f>>n;
for(i=1;i<=n;i++)
{f>>x;
if(x>max) max=x;
if(x<min) min=x;}
cout<<max<<" "<<min;}
```

Varianta PASCAL

```
var n,i :longint; val, min, max:
byte; f:text;
begin
min:=10;max:=-1;
assign(f,'numere.txt'); reset(f);
read (f,n);
for i:=1 to n do begin
read(f,val); if val<min then
min:=val;
if val>max then
max:=val; end;
write (max,' ',min);
close(f);end.
```

b) Neintensiv.S-a citit fiecare din cele n cifre fără le memora, s-a comparat cu **max** și **min**, toate valorile și s-au reținut valorile extreme. Prelucrarea s-a facut odatăcu citirea (O

singură parcurgere a celor n numere) – complexitate $O(n)$.

Varianta <42>

ȘUBIECTUL I

2. a) 5

b) Algoritmul calculează **cmmdc**. Cerința b) pentru $y=20$ și rezultatul să fie 10 ar rezulta $x=90$.

c) citește x, y
 executa
 $z=x\%y$
 $x=y$
 $y=z$
 pana cand $y\leq 0$;
 scrie x

d) Varianta C/C++

```
#include<iostream.h>
void main()
{int x,y,z;
cout<<"x=";<<cin>>x;
cout<<"y=";<<cin>>y;
while(y>0) {z=x%y; x=y; y=z; }
cout<<"x="<<x;}
```

Varianta PASCAL

```
var x,y,z:integer;
begin
write('x=');read(x);
write('y=');read(y);
while y>0 do begin
z:=x mod y;
x:=y;y:=z; end;
write('x=',x);
```

ȘUBIECTUL II

2. a) pentru varianta intensiv, respectiv a pentru neintensiv

3. 6 (1-5-3-2-7-4-8)

4. 5

5. Varianta C/C++

```
#include<iostream.h>
#include<string.h>
void main()
{char a[40];int i;
cin.get(a,40);
for(i=0;i<strlen(a);i++)
if(a[i]=='a' || a[i]=='e' || a[i]=='i' || a[i]=='o' || a[i]=='u') cout<<a[i]<<" ";}
```

Varianta PASCAL

```
var sir: string[40]; i, lung
:integer;
begin
write ('Sirul este: ');
read(sir);
lung:=length(sir);
for i:=1 to lung do
if sir[i] in
['a','e','i','o','u'] then
write(sir[i], ' ');
end.
```

SUBIECTUL III

1. a

2. 6 și 96

3.

Varianta C/C++

```
long P(int a[100],int n,int k)
{long max=0,s=0;
int i,j;
```

Varianta PASCAL

```
Type vector=array[1..100] of
integer;
Function
P(a:vector,n:integer,k:integer);
```

```

while(k>0)
{max=-10000;
for(i=1;i<=n;i++)
if(a[i]>max) {max=a[i];j=i;}
s+=a[j];
a[j]=-10000; k--;}
return s;
}

```

4. a)

Pentru intensiv:

Varianta C/C++

```

#include<fstream.h>

fstream f("numere.txt",
ios::in);
int a[100];
void main()
{int n,x;
f>>n;
for(int
i=1;i<=n;i++){f>>x;a[x]++;}
for(i=1;i<=99;i++)
if(a[i]==1) cout<<i<<" ";}

```

b) Pentru cele **n**numere ($n < 100000$) nu s-a utilizat un vector pentru stocare ci s-a utilizat un vector cu 100 elemente care a stocat numarul de apariții pentru fiecare valoare (0-99) prelucrarea s-a făcut odată cu citirea, pentru afișare s-a parcurs vectorul de 100 elemente rezultând complexitate $O(n)$.

Pentru neintensiv:

Varianta C/C++

```

#include<fstream.h>
fstream f("numere.txt", ios::in);
void main()
{int n,x1,x2,i,ok=1;
f>>n; f>>x1;
for(i=1;i<n;i++) {f>>x2;
if(x2<=x1) ok=0;

if(ok) cout<<"DA"; else cout<<"NU";}
}

```

b) Pentru cele **n**numere ($n < 100000$) nu s-a utilizat un vector pentru stocare ci s-a citit fiecare valoare și s-a comparat cu valoarea precedentă determinându-se dacă valorile sunt în ordine strict crescătoare. Rezultă complexitate $O(n)$.

```

Var i,j,aux:integer;
s:longint;
begin
for i:=1 to n-1 do
for j:=i+1 to n do
if a[i]<a[j] then
begin
aux:=a[i];a[i]:=a[j];a[j]:=aux;
end;
s:=0;
for i:=1 to k do s:=s+a[i];
P:=s;
end;

```

Varianta PASCAL

```

var n,i :longint; val: byte;
v :array [0..99] of byte;
f:text;
begin
assign(f,'numere.txt'); reset(f);
read (f,n);
for i:=1 to n do begin
read(f,val); inc(v[val]); end;
for i:=0 to 99 do
if v[i]=1 then write (i,' ');
close(f); end.

```

Varianta PASCAL

```

var n,i :longint; val, uval:
integer;
ok: boolean;
begin
assign(f,'numere.txt'); reset(f);
read (f,n); uval=-1; ok:=true;
for i:=1 to n do begin
read(f,val);
if val<=uval then ok:=false
else uval:=val;
end;
if ok=true then write ('DA') else
write('NU');
close(f); end.

```

Varianta <43 intensiv>

SUBIECTUL I

1. a
- 2.

b) 13, 39, 65, 91

c)

```
citeste x,y
repetă
    dacă x>y atunci x<-x%y
        altfel y<-y%x
sfarsit dacă
pana cand x*y=0
```

d)

Varianta C/C++

```
#include <iostream.h>
void main()
{int x,y;
  cout<<"x=";cin>>x;
  cout<<"y=";cin>>y;
  while(x*y !=0)
    if(x>y) x=x%y; else y=y%x;
  cout<<x+y;
}
```

Varianta PASCAL

```
program d2;
var x,y:integer;
begin
  write('x=');readln(x);
  write('y=');readln(y);
  while(x*y<>0) do
    If x>y then x:=x mod y
      else y:=y mod x;
  writeln(x+y);
end.
```

SUBIECTUL II

1. a
2. a
3. rădăcina este 4; arborele are 5 frunze
4. inmatrica
- 5.

Varianta C/C++

```
#include<iostream.h>
void main()
{int n,i,j, mat[24][24];
  cout<<"n=";cin>>n;

  for (i=1 ;i<=n ;i++) mat[i][i]=2;
  for (i=1;i<=n-1;i++)
    for(j=i+1;j<=n;j++) mat[i][j]=1;
  for(i=2;i<=n;i++)
    for(j=1;j<=i-1;j++) mat[i][j]=3;

  for(i=1;i<=n;i++)
  {cout<<endl;
    for(j=1;j<=n;j++)
      cout<<mat[i][j]<<" ";
  }
}
```

Varianta PASCAL

```
program cinci;
var n,i,j:integer;
    mat:array[1..23,1..23] of
integer;
begin
  write('n=');readln(n);
  for i:=1 to n do mat[i,i]:=2;
  for i:=1 to n-1 do
    for j:=i+1 to n do mat[i,j]:=1;
  for i:=2 to n do
    for j:=1 to i-1 do mat[i,j]:=3;
  for i:=1 to n do
    begin
      writeln;
      for j:=1 to n do write(mat[i,j],
');
    end;
end.
```

SUBIECTUL III

1. a
2. f(3)=6, f(10)=20

3.

Se citește n , apoi se citesc cele n numere succesiv în variabila x . Fiecare număr este verificat; dacă este impar, atunci se adaugă la suma s .

Varianta C/C++

```
#include<iostream.h>
void main()
{int n,x,i; long s=0;
  cout<<"n=";cin>>n;}
  for (i=1;i<=n;i++)
    {cout<<"dati un numar=";cin>>x;
     if (x%2!=0) s=s+x;
    }
  cout<<s;
}
```

Varianta PASCAL

```
program trei;
var n,i,x:integer;s:longint;
begin
write('n=');readln(n);
s:=0;
for i:=1 to n do
begin
write('dati un numar=');readln(x);
if x mod 2 <>0 then s:=s+x;
end;
writeln(s);
end.
```

4.

a) Varianta C/C++

```
#include<fstream.h>
unsigned long cifre[10];
void main()
{unsigned long n,i,j;
int k,cif;
ifstream f("numere.txt");
f>>n;
for (i=1;i<=n;i++)
{f>>cif;
cifre[cif]++;

f}close();
for (k=9;k>=0;k--)
for (j=1;j<=cifre[k];j++)
cout<<k;
}
```

Varianta PASCAL

```
program patru;
var n,i,j:longint;k,cif:byte;
cifre:array[0..9] of longint;
f:text;
begin
for i:=0 to 9 do cifre[i]:=0;
assign (f,'numere.txt');
reset(f);readln(f,n);
for i:=1 to n do
begin
read(f,cif);cifre[cif]:=cifre[cif]+1;
end;
close(f);
for k:=9 downto 0 do
for j:=1 to cifre[k] do
write(k);
end.
```

b)

În vectorul *cifre* contorizează numărul de apariții al fiecărei cifre *cif*. În final se parcurge vectorul *cifre* în ordinea descrescătoare a indicilor (de la 9 la 0) și se afișează fiecare cifră de câte ori apare.

Varianta <43 neintensiv>

SUBIECTUL I

1. a

2.

a) 5

b) 13

c)

```
citeste x,y
repetă
daca x>y atunci x<-x*y
sfarsit dacă altfel y<-y*x
pana cand x*y=0
```

d)

Varianta C/C++

```
#include <iostream.h>
void main()
{int x,y;

    cout<<"x=";<<cin>>x;
    while(x*y !=0)
        if(x>y) x=x%y; else y=y%x;
    cout<<x+y;
}
```

Varianta PASCAL

```
program d2;
var x,y:integer;
begin

write('x=');readln(x);
while(x*y<>0) do
    If x>y then x:=x mod y
        else y:=y mod x;
writeln(x+y);
end.
```

SUBIECTUL II

1. a
2. a
3. rădăcina este 4; arborele are 5 frunze
4. 11
- 5.

Varianta C/C++

```
#include<iostream.h>
void main()
{int n,i,j, mat[24][24];
    cout<<"n=";<<cin>>n;

for (i=1 ;i<=n ;i++) mat[i][i]=2;
    for (i=1;i<=n-1;i++)
        for(j=i+1;j<=n;j++) mat[i][j]=1;
    for(i=2;i<=n;i++)
        for(j=1;j<=i-1;j++) mat[i][j]=3;

for(i=1;i<=n;i++)
    {cout<<endl;
        for(j=1;j<=n;j++)
            cout<<mat[i][j]<<" ";
    }
}
```

Varianta PASCAL

```
program cinci;
var n,i,j:integer;
    mat:array[1..23,1..23] of
integer;
begin
write('n=');readln(n);

for i:=1 to n do mat[i,i]:=2;
for i:=1 to n-1 do
    for j:=i+1 to n do mat[i,j]:=1;
for i:=2 to n do
    for j:=1 to i-1 do mat[i,j]:=3;

for i:=1 to n do
    begin
        writeln;
        for j:=1 to n do write(mat[i,j],'
');
    end;
end.
```

SUBIECTUL III

1. a
2. f(3)=6, f(10)=20
- 3.

Se citește **n** de la tastatura, apoi se citesc cele **n** numere succesiv în variabila x. Fiecare număr este verificat; dacă este impar, atunci se adaugă la suma s.

Varianta C/C++

```
long P(int x[100],int n)
{
int i; long s=0;
for (i=1;i<=n;i++)
    if (x[i]%2!=0) s=s+x[i];
return s;
}
```

Varianta PASCAL

```
Type vector=array[1..100] of integer;
Function
f(x:vector,n:integer):longint;
var i:integer;s:longint;
begin
s:=0;
for i:=1 to n do
    if x[i] mod 2 <>0 then s:=s+x[i];
```

4.

a)

Varianta C/C++

```
#include<fstream.h>
void main()
{ifstream f("numere.txt");
 unsigned long n,i, nrapar=1;
 f>>n;
 int x,cifmax=0;
 for (i=1;i<=n;i++)
 {f>>x;
  if (x==cifmax) nrapar++;
  else if (x>cifmax){cifmax=x;
                    nrapar=1;
  }
 }
 f.close();
 cout<<cifmax<<" "<<nrapar;
 }
```

```
P:=s;
end.
```

Varianta PASCAL

```
program patru;
var l,n,hrapaf:longint;
    x,cifmax:byte;
    f:text;
begin
 assign (f,'numere.txt');reset(f);
 readln(f,n);
 nrapar:=1;cifmax:=0;
 for i:=1 to n do
 begin
  read(f,x);
  if x=maxcif then nrapar:=nrapar+1
  else if x>cifmax then begin
                    cifmax:=x;
                    nrapar:=1;
  end;
 end;
 close(f);
 write(cifmax,' ',nrapar);
end.
```

b) Variabila *nrapar* ține numărul de apariții curente pentru cifra maximă *cifmax* întâlnită în fișier. Dacă în fișier se întâlnește o cifră *x* identică cu *cifmax*, atunci se incrementează contorul *nrapar*. Dacă în fișier se întâlnește o cifră *x* mai mare decât cifra maximă *cifmax*, atunci se actualizează cele două variabile.

Varianta <44 intensiv>

SUBIECTUL I

1. a

2.

a) 555

b) Toate numerele de 3 cifre care au cifra unităților 8, cu excepția lui 108 (care dă y=111). Adică: 118, 128, 138, 148, ..., 998

c)

citeste x

```
y<=0
repetă
  y=y*10+9-x%10
pana cand x<=y
scrie y
```

d)

Varianta C/C++

```
#include <iostream.h>
void main()
{unsigned long x,y;
 cout<<"x=";<<cin>>x;
 y=0;

 while(x>y) y=y*10+9-x%10 ;
 cout<<y;
 }
```

Varianta PASCAL

```
program unud;
var x,y:longint;
begin
 write('x=');readln(x);
 y:=0;

 while(x>y) do y:=y*10+9-x mod 10;
end.
```


SUBIECTUL II

1. a

2. a

3. 2, 6, 7

4. jogp

5. Explicații

Plasăm sub diagonala secundară, pe fiecare linie și coloană, șirul de numere 1, 2, 3, ..., n, începând cu valori de 1 pe diagonală. Iar deasupra diagonalei plasăm numai valori de 1. Ideea este că fiecare linie și coloană este împărțită în două părți de diagonala respectivă. Asemănător se poate proceda folosind diagonala principală.

Varianta C/C++

```
#include<iostream.h>
void main()
{int n,i,j,k,mat[24][24];
 cout<<"n=";cin>>n;
 for(i=1;i<=n;i++)
  {k=0;
   for (j=n-i+1;j<=n;j++)
    {k++;mat[i][j]=k;}
  }
 for (i=1;i<=n-1;i++)
  for (j=1;j<=n-i+1;j++) mat[i][j]=1;
 for (i=1;i<=n;i++)
  {cout<<endl;
   for(j=1;j<=n;j++)
    cout<<mat[i][j]<<" ";
  }
}
```

Varianta PASCAL

```
program cinci;
var n,i,j,k:integer;
    mat:array[1..23,1..23] of
    integer;
begin
 write('n=');readln(n);
 for i:=1 to n do
  begin
   k:=0;
   for j:=n-i+1 to n do
    begin
     k:=k+1;mat[i,j]:=k;
    end;
  end;
 for i:=1 to n-1 do
  for j:=1 to n-i do mat[i,j]:=1;
 for i:=1 to n do
  writeln;
  for j:=1 to n do write(mat[i,j],
  ');
 end;
end.
```

SUBIECTUL III

1. a

2. 7

3.

Pentru fiecare număr x citit, se află radicalul sau ca număr natural în y . Apoi, prin înmulțiri repetate $y \cdot y$, se verifică dacă este pătrat perfect; în acest caz se mărește conținutul n de p pătrate perfecte.

Varianta C/C++

```
#include<iostream.h>
#include<math.h>
void main()
{int n,x,i,nrp=0;
 cout<<"n=";cin>>n;
 for (i=1;i<=n;i++)
  {cin>>x;
   if (sqrt(x)==floor(sqrt(x)))
    nrp++;
  }
 cout<<nrp;
}
```

Varianta PASCAL

```
program cinci;
var n,i,x,nrp:integer;
begin
 write('n=');readln(n);
 nrp:=0;
 for i:=1 to n do
  begin
   read(x);
   if (sqrt(x)=round(sqrt(x))) then
    nrp:=nrp+1;
  end;
 write(nrp);
end.
```

4.

a)

Varianta C/C++

```
#include<fstream.h>

int cifre[10];
void main()
{unsigned long n,i;
int x,cif;
ifstream f("numere.txt");
f>>n;
for (i=1;i<=n;i++)
    {f>>x;
    while(x)
    {cif=x%10;cifre[cif]=1;x/=10;}
    }
f.close();
for (i=0;i<=9;i++)
    if (cifre[i]==1) cout<<i<<" ";
```

end.

Varianta PASCAL

```
program patru;
var n,i:longint;x:integer;cif:byte;
    cifre:array[0..9] of byte;
    f:text;
begin
for i:=0 to 9 do cifre[i]:=0;
assign (f,'numere.txt');reset(f);
readln(f,n);
for i:=1 to n do
begin
read(f,x);
while (x>0) do
begin
cif:=x mod 10;cifre[cif]:=1;
x:=x div 10;
end;
end;
close(f);
for i:=0 to 9 do
if cifre[i]=1 then write(i,' ');
end.
```

b)

In vectorul *cifre* se marchează cu 1 apariția fiecărei cifre *cif* din fiecare număr *x* (citit din fișier). Indexul (poziția) din vectorul *cifre* reprezintă valoarea unei cifre. In final se parcurge vectorul *cifre* și se afișează acei indici și pentru care elementul este marcat cu 1.

Varianta <44 neintensiv>

SUBIECTUL I

1. a

2.

a) 555

b) Toate numerele de 3 cifre care au cifra unităților 8, cu excepția lui 108 (care dă y=111). Adică: 118, 128, 138, 148, ..., 998

c)

citeste x

```
y<=0
repetă
    y=y*10+9-x%10
pana cand x<=y
scrie y
```

d)

Varianta C/C++

```
#include <iostream.h>
void main()
{unsigned long x,y;
cout<<"x=";cin>>x;
y=0;

while(x>y) y=y*10+9-x%10 ;
cout<<y;
}
```

Varianta PASCAL

```
program unud;
var x,y:longint;
begin
write('x=');readln(x);
y:=0;

while(x>y) do y:=y*10+9-x mod 10;
end.
```

SUBIECTUL II

1. a

2. d

3. 2, 6, 7

4. fo

5. Explicații

Plasăm sub diagonala secundară, pe fiecare linie și coloană, șirul de numere 1, 2, 3, ..., n, începând cu valori de 1 pe diagonală. Iar deasupra diagonalei plasăm numai valori de 1. Ideea este că fiecare linie și coloană este împărțită în două părți de diagonala respectivă. Asemănător se poate proceda folosind diagonala principală.

Varianta C/C++

```
#include<iostream.h>
void main()
{int n,i,j,k,mat[24][24];
 cout<<"n=";cin>>n;
 for(i=1;i<=n;i++)
  {k=0;
   for (j=n-i+1;j<=n;j++)
    {k++;mat[i][j]=k;}
  }
 for (i=1;i<=n-1;i++)
  for (j=1;j<n-i+1;j++) mat[i][j]=1;
 for (i=1;i<=n;i++)
  {cout<<endl;
   for(j=1;j<=n;j++)
    cout<<mat[i][j]<<" ";
  }
}
```

Varianta PASCAL

```
program cinci;
var n,i,j,k:integer;
    mat:array[1..23,1..23] of
integer;
begin
write('n=');readln(n);
for i:=1 to n do
begin
k:=0;
for j:=n-i+1 to n do
begin
k:=k+1;mat[i,j]:=k;
end;
end;
for i:=1 to n-1 do
for j:=1 to n-i do mat[i,j]:=1;
for i:=1 to n do
writeLn;
for j:=1 to n do write(mat[i,j],
');
end;
end.
```

SUBIECTUL III

1. a

2. 7

3.

Pentru fiecare număr x citit, se află radicalul sau ca număr natural în y . Apoi, prin înmulțiri repetate $y*y$, se verifică dacă este atins x ; în acest caz se mărește contorul n de patru perfect.

Varianta C/C++

```
#include<iostream.h>
#include<math.h>
void main()
{int n,x,y,i,nrp=0;
 cout<<"n=";cin>>n;
 for (i=1;i<=n;i++)
  {cin>>x;
   if (sqrt(x)==floor(sqrt(x)))
    nrp++;
  }
 cout<<nrp;
}
```

Varianta PASCAL

```
program cinci;
var n,i,x,nrp:integer;
begin
write('n=');readln(n);
nrp:=0;
for i:=1 to n do
begin
read(x);
if (sqrt(x)=round(sqrt(x))) then
nrp:=nrp+1;
end;
write(nrp);
```

4.

a)

Varianta C/C++

```
#include<fstream.h>

unsigned long nr[100];
void main()
{unsigned long n,i;
int x;
ifstream f("numere.txt");
f>>n;
for (i=1;i<=n;i++)
    {f>>x;
    nr[x]++;
    }
f.close();
for (i=0;i<=99;i++)
    if (nr[i]>=2) cout<<i<<" ";
}
```

b)

Vectorul nr memorează numărul de apariții $nr[i]$ al fiecărui număr i de două cifre din intervalul $0 \dots 99$. În final, se afișează numerele i care au $nr[i] \geq 2$, adică care au cel puțin două apariții.

end.

Varianta PASCAL

```
program patru;
var n,i:longint;x:byte;
    nf:array[0..99] of longint;
    f:text;
begin
for i:=0 to 99 do nr[i]:=0;
assign (f,'numere.txt');
reset(f);readln(f,n);
for i:=1 to n do
begin
read(f,x);
nr[x]:=nr[x]+1;
end;
close(f);
for i:=0 to 99 do
if nr[i]>=2 then write(i,' ');
end.
```

Varianta <45 intensiv>

SUBIECTUL I

1. a

2. a) 9

b) 38

c)

```
citeste x,y
z<-1
t<-0
repetă
    dacă x%z=y atunci t←z
    sfârșit dacă
pană când x<z
```

d)

Varianta C/C++

```
#include<iostream.h>
void main()
{int x,y,z,t;
cin>>x>>y;
z=1;t=0;
while (x>=z)
    {if (x%z==y) t=z;
    z++;
    }
cout<<t;
```

Varianta PASCAL

```
program doid;
var x,y,z,t:integer;
begin
read(x,y);
z:=1;t:=0;
while x>=z do
begin
if x mod z=y then t:=z;
z:=z+1;
end;
write(t);
end.
```

SUBIECTUL II

1. a

2. a

3. Descendenții direcți ai rădăcinii sunt 1 și 7. Arborele are 4 frunze.

4. Cel mai mic element din linia a doua are valoarea -3 și este situat în coloana 5.

5. Explicații

Din șirul auxiliar `yse` șterge pe rând câte o literă și se afișează șirul rezultat.

Varianta C/C++

```
#include <iostream.h>
#include <string.h>
void main()
{char sir[40],y[40];
 int i;
 cin.getline(sir,40,'\n');
 int n=strlen(sir);
 for (i=0;i<n;i++)
 {strcpy(y,sir);
  strcpy(y+i,y+i+1);
  cout<<y<<endl;
 }
}
```

Varianta PASCAL

```
program doicinci;
var sir,y:string[40];
    i,n:integer;
begin
read(sir);n:=length(sir);
for i:=1 to n do
begin
y:=sir;
delete(y,i,1);
writeln(y);
end;
end.
```

SUBIECTUL III

1. a

2. $f(4)=3$ și $f(11)=5$

3.

Cele n numere reale se citesc în vectorul v . Fiecare număr $v[i]$ încă neverificat ($mark[i]=0$) este marcat ($mark[i]=1$), iar apoi se caută de câte ori mai apare el în restul vectorului (pozițiile $i+1 \dots n-1$). Dacă apare o singură dată în tot vectorul ($nr=1$), atunci este afișat.

Varianta C/C++

```
#include<iostream.h>
int mark[99];
void main()
{int n,i,j; float v[99];
 cout<<"n=";cin>>n;
 for (i=0;i<n;i++)
 {cout<<"numar=";cin>>v[i];
 }
 for (i=0;i<n-1;i++)
 if (!(mark[i]))
 {mark[i]=1;
  int nr=1;
  for (j=i+1;j<n;j++)
   if (v[j]==v[i])
    {nr++;mark[j]=1;}
  if (nr==1) cout<<v[i]<<" ";
 }
}
```

Varianta PASCAL

```
program trei;
var n,i,j,nr:integer;
    v:array [0..99] of real;
    mark:array[0..99] of byte;
begin
write('n=');readln(n);
for i:=0 to n-1 do read(v[i]);
for i:=0 to n-1 do mark[i]:=0;
for i:=0 to n-2 do
if mark[i]=0 then
begin
mark[i]:=1;nr:=1;
for j:=i+1 to n-1 do
if (v[i]=v[j])
then begin nr:=nr+1;
mark[j]:=1;
end;
if nr=1 then write(v[i]:3:3,' ');
end;
end.
```

4.

a)

Varianta C/C++

```
#include<fstream.h>
unsigned long nr[100];

void main()
{unsigned long n,i,j;
int x;
ifstream f("numere.txt");
f>>n;
for (i=1;i<=n;i++)
    {f>>x;
    nr[x]++;
    }
f.close();
for (i=0;i<=99;i++)
    for (j=1;j<=nr[i];j++)
        cout<<i<<" ";
}
```

Varianta PASCAL

```
program patru;
var n,i,j:longint;x:byte;

    nr:array[0..99] of longint;
    f:text;
begin
for i:=0 to 99 do nr[i]:=0;
assign (f,'numere.txt');
reset(f);readln(f,n);
for i:=1 to n do
    begin
    read(f,x);
    nr[x]:=nr[x]+1;
    end;
close(f);
for i:=0 to 99 do
    for j:=1 to nr[i] do
        write(i,' ');
    end.
```

b)

Numerele formate din cel mult 2 cifre sunt 0 ...99, deci este suficient un vector *nrcu* 100 elemente. Fiecare element *nr[i]* memoreaza de câte ori apare un numar *i*. In final sunt afișate în ordine crescătoare (și de câte ori apare fiecare) numerele de la 0 la 99.

Varianta <45 neintensiv>

SUBIECTUL I

1. a

2. a) 9

b) 38

c)

```
citeste x,y
z<-1
t<-0
repetă
    dacă x%z=y atunci t←z
    sfarsit dacă

pana cand x<z
```

d)

Varianta C/C++

```
#include<iostream.h>
void main()
{int x,y,z,t;
cin>>x>>y;
z=1;t=0;
while (x>=z)
    {if (x%z==y) t=z;
    z++;
    }
cout<<t;
```

Varianta PASCAL

```
program doid;
var x,y,z,t:integer;
begin
read(x,y);
z:=1;t:=0;
while x>=z do
    begin
    if x mod z=y then t:=z;
    z:=z+1;
    end;
write(t);
end.
```

SUBIECTUL II

1. a

2. a

3. Descendenții direcți ai rădăcinii sunt 1 și 7. Arborele are 4 frunze.

4. Cel mai mic element din linia a doua are valoarea 1 și este situat în coloana 1.

5. Explicații

Din șirul auxiliar `yse` șterge pe rând câte o literă și se afișează șirul rezultat.

Varianta C/C++

```
#include <iostream.h>
#include <string.h>
void main()
{char sir[40],y[40];
 int i;
 cin.getline(sir,40,'\n');
 int n=strlen(sir);
 for (i=0;i<n;i++)
 {strcpy(y,sir);
  strcpy(y+i,y+i+1);
  cout<<y<<endl;
 }
}
```

Varianta PASCAL

```
program doicinci;
var sir,y:string[40];
    i,n:integer;
begin
read(sir);n:=length(sir);
for i:=1 to n do
begin
y:=sir;
delete(y,i,1);
writeln(y);
end;
end.
```

SUBIECTUL III

1. a

2. $f(4)=3$ și $f(11)=5$

3. Explicații

Cele n numere reale se citesc în vectorul v . Fiecare număr $v[i]$ încã neverificat ($mark[i]=0$) este marcat ($mark[i]=1$), iar apoi se cautã de câte ori mai apare el în restul vectorului (pozițiile $i+1 \dots n-1$). Dacă apare o singurã datã în tot vectorul ($nr=1$), atunci este afișat.

Varianta C/C++

```
#include<iostream.h>
int mark[99];
void main()
{int n,i,j; float v[99];
 cout<<"n=";cin>>n;
 for (i=0;i<n;i++)
 {cout<<"numar=";cin>>v[i];
 }

for (i=0;i<n;i++)
 {mark[i]=1;
  int nr=1;
  for (j=i+1;j<n;j++)
  if (v[j]==v[i])
  {nr++;mark[j]=1;}
  if (nr==1) cout<<v[i]<<" ";
 }
}
```

Varianta PASCAL

```
program trei;
var n,i,j,nr:integer;
    v:array [0..99] of real;
    mark:array[0..99] of byte;
begin
write('n=');readln(n);
for i:=0 to n-1 do read(v[i]);
for i:=0 to n-1 do mark[i]:=0;
for i:=0 to n-2 do
if mark[i]=0 then
begin
mark[i]:=1;nr:=1;
for j:=i+1 to n-1 do
if (v[i]=v[j])
then begin nr:=nr+1;
mark[j]:=1;
end;
if nr=1 then write(v[i]:3:3,' ');
end;
end.
```

4.

a)

Varianta C/C++

Varianta PASCAL

```
#include<fstream.h>
unsigned long nr[100];
void main()
{unsigned long n,i,j;
int x;
ifstream f("numere.txt");
f>>n;
for (i=1;i<=n;i++)
{f>>x;
nr[x]++;
}
f.close();
for (i=0;i<=99;i++)
for(j=1;j<=nr[i];j++)
cout<<i<<" ";
}
```

```
program patru;
var n,i,j:longint;x:byte;
nr:array[0..99] of longint;
f:text;
begin
for i:=0 to 99 do nr[i]:=0;
assign (f,'numere.txt');
reset(f);readln(f,n);
for i:=1 to n do
begin
read(f,x);
nr[x]:=nr[x]+1;
end;
close(f);
for i:=0 to 99 do
for j:=1 to nr[i] do
write(i,' ');
end.
```

b) Numerele formate din cel mult 2 cifre sunt 0 ...99, deci este suficient un vector nr cu 100 elemente. Fiecare element $nr[i]$ memorează câte ori apare un număr i . În final sunt afișate în ordine crescătoare (și de câte ori apare fiecare) numerele de la 0 la 99.

Varianta <46 intensiv>

SUBIECTUL I

1. c

2.

a) 1

b) 60

c)

```
citeste n
s<-0
nr<-0
repetă
daca n%2=0 atunci s<-s*10+n%10
sfarsit daca
n<-[n/10]
pana cand n=0
daca s≠0 atunci nr←-1
sfarsit daca
scrie nr
```

d)

Varianta C/C++

```
#include<iostream.h>
void main()
{int n,s=0,nr=0;
cout<<"n=";>>n;
while (n!=0)
{if (n%2==0) s=s*10+n%10;
n=n/10;
}
if (s!=0) nr=1;
cout<<nr;
}
```

Varianta PASCAL

```
program doi;
var n,s,nr:integer;
begin
write('n=');readln(n);
s:=0;nr:=0;
while n<>0 do
begin
if n mod 2=0 then s:=s*10+n mod 10;
n:=n div 10;
end;
if s<>0 then nr:=1;
write(nr);
end.
```


SUBIECTUL II

1. b

2. a

3. 6

4. 13

5. Explicații

Sirul de lungime n este parcurs caracter cu caracter. Pentru fiecare vocala se afla codul ASCII si se mareste cu o unitate. In sir, in locul vocalei se pune caracterul ce are noul cod ASCII determinat.

Varianta C/C++

```
#include<iostream.h>
#include<string.h>
void main()
{char sir[100];
cout<<"sir=";cin.getline(sir,100,'\n');
int n=strlen(sir);
for (int i=0;i<n;i++)
if ((sir[i]=='a')||(sir[i]=='e')
    ||(sir[i]=='i')||(sir[i]=='o')
    ||(sir[i]=='u'))
{int cod=sir[i];cod++;
sir[i]=cod;
}
cout<< sir;
}
```

Varianta PASCAL

```
program cinci;
var
sir:string[100];n,i,cod:integer;
begin
write('sir=');readln(sir);
n:=length(sir);
for i:=1 to n do
if sir[i] in
['a','e','i','o','u']
then
begin
cod:=ord(sir[i]);cod:=cod+1;
sir[i]:=char(cod);
end;
writeln(sir);
end.
```

SUBIECTUL III

1. c

2. 100

3. Explicații

Variabilele folosite au urmatoarele semnificatii: n -cate numere sunt in fisier; x -contine pe rand fiecare numar din fisier; $pozv$ -pozitia curenta in vectorul de afisare v ; nrd -cati divizori are fiecare numar x ; d -contor pentru divizorii posibili. Pentru fiecare numar citit din fisier in x , se initializeaza nrd cu valoarea 2(orice numar are cel puțin 2 divizori) si se cauta divizorii incepand cu 2 si pana la jumatarea numarului x . Daca nrd ajunge cel puțin la valoarea k , atunci numarul x se depune in v pe pozitia $pozv$.

Varianta C/C++

```
#include<fstream.h>
void main()
{int n,i,x,k,v[100];
cout<<"k=";cin>>k;
ifstream f("bac.txt");
f>>n;int pozv=-1;
for(i=1;i<=n;i++)
{f>>x;
int nrd=2;
for (int d=2;d<=x/2 &&
nrd<k;d++)
if (x%d==0) nrd++;
if (nrd>=k) {pozv++;v[pozv]=x;}
}
f.close();
for(i=0;i<=pozv;i++) cout<<v[i]<<"
";
```

Varianta PASCAL

```
program trei;
var n,pozv,i,x,nrd,d,k:integer;
v:array[1..100] of integer;
f:text;
begin
write('k=');readln(k);
assign(f,'bac.txt');reset(f);
readln(f,n);pozv:=0;
for i:=1 to n do
begin
read(f,x);nrd:=2;d:=2;
while(d<=x/2) and (nrd<k) do
begin
if x mod d=0 then nrd:=nrd+1;
d:=d+1;
end;
if nrd>=k then begin
```

}

```

        pozv:=pozv+1;
        v[pozv]:=x;
        end;
    end;
close(f);
for i:=1 to pozv do write(v[i], ' ');
end.

```

4.

a)

Varianta C/C++

```

void cifre(unsigned long nr,int
&nc, int &sc)

```

b)

```

int subp(unsigned long x)
{int nc,sc;
  cifre(x,nc,sc);

  int media=sc/nc;
  int ok=0;
  while (!ok && x)
  { int cif=x%10;
    ok=cif==media;
    x/=10;
  }
  return ok;
}

```

Varianta PASCAL

```

procedure cifre(x:longint;
var nc,sc:integer);

```

```

function subp(x:longint):integer;
var nc,sc,media,cif,ok:integer;
begin
  cifre(x,nc,sc);
  media:=sc div nc;ok:=0;
  while (ok=0) and (x>0) do
  begin
    cif:=x mod 10;
    if cif=media then ok:=1;
    x:=x div 10;
  end;
  subprog:=ok;
end;

```

SUBIECTUL I

1. c

2.

a) 1

b) 60

c)

```

citeste n
s<-0
nr<-0
repete
  daca n%2=0 atunci s<-s*10+n%10
  sfarsit daca
  n<-[n/10]
pana cand n=0
daca s≠0 atunci nr←-1
sfarsit daca
scrie nr

```

d)

Varianta C/C++

```

#include<iostream.h>
void main()
{int n,s=0,nr=0;
cout<<"n=";<<cin>>n;
while (n!=0)
  {if (n%2==0) s=s*10+n%10;

```

Varianta PASCAL

```

program doi;
var n,s,nr:integer;
begin
write('n=');readln(n);
s:=0;nr:=0;
while n<>0 do

```

Varianta <46 neintensiv>

```

    n=n/10;
}
if (s!=0) nr=1;
cout<<nr;
}

```

```

begin
if n mod 2=0 then s:=s*10+n mod 10;
n:=n div 10;
end;
if s<>0 then nr:=1;
write(nr);
end.

```

SUBIECTUL II

1. b

2. a

3. 6

4. 1

5. Explicații

Sirul de lungime n este parcurs caracter cu caracter. Pentru fiecare vocala se afla codul ASCII si se mareste cu o unitate. In sir, in locul vocalei se pune caracterul ce are noul cod

ASCII determinat.

Varianta C/C++

```

#include<iostream.h>
#include<string.h>
void main()
{char sir[100];
cout<<"sir=";cin.getline(sir,100,'\n');
int n=strlen(sir);
for (int i=0;i<n;i++)
if ((sir[i]=='a')||(sir[i]=='e')
|| (sir[i]=='i')||(sir[i]=='o')
|| (sir[i]=='u'))
{int cod=sir[i];cod++;
sir[i]=cod;
}
cout<<sir;
}

```

Varianta PASCAL

```

program cinci;
var
sir:string[100];n,i,cod:integer;
begin
write('sir=');readln(sir);
n:=length(sir);
for i:=1 to n do
if sir[i] in
['a','e','i','o','u']
then
begin
cod:=ord(sir[i]);cod:=cod+1;
sir[i]:=char(cod);
end;
writeln(sir);
end.

```

SUBIECTUL III

1. c

2. $f(7)=56$, $f(100)=10100$

3. Explicații

Variabilele folosite au urmatoarele semnificatii: n -cate numere sunt in fisier; x -contine pe rand fiecare numar din fisier; $pozv$ -pozitia curenta in vectorul de afisare v ; nrd -cati divizori are fiecare numar x ; d -contor pentru divizorii posibili. Pentru fiecare numar citit din fisier in x , se initializeaza nrd cu valoarea 2(orice numar are cel putin 2 divizori) si se cauta divizorii incepand cu 2 si pana la jumatatea numarului x . Daca nrd ajunge cel putin la valoarea k , atunci numarul x se depune in v pe pozitia $pozv$.

Varianta C/C++

```

#include<fstream.h>
void main()
{int n,i,x,k,v[100];
cout<<"k=";cin>>k;
ifstream f("bac.txt");
f>>n;int pozv=-1;
for(i=1;i<=n;i++)
{f>>x;

```

Varianta PASCAL

```

program trei;
var n,pozv,i,x,nrd,d,k:integer;
v:array[1..100] of integer;
f:text;
begin
write('k=');readln(k);
assign(f,'bac.txt');reset(f);
readln(f,n);pozv:=0;

```

```

int nrd=2;
for (int d=2;d<=x/2 &&
nrd<k;d++)
    if (x%d==0) nrd++;
    if (nrd>=k) {pozv++;v[pozv]=x;}
}
f.close();
for(i=0;i<=pozv;i++) cout<<v[i]<<"
";
}

```

```

for i:=1 to n do
begin
read(f,x);nrd:=2;d:=2;
while(d<=x/2) and (nrd<k) do
begin
if x mod d=0 then nrd:=nrd+1;
d:=d+1;
end;
if nrd>=k then begin
pozv:=pozv+1;
v[pozv]:=x;
end;

end;
close(f);
for i:=1 to pozv do write(v[i],' ');
end.

```

4.

a)

Varianta C/C++

```

void cifre(unsigned long nr,int
&nc, int &sc)

```

b)

```

int subp(unsigned long x)
{int nc,sc;
cifre(x,nc,sc);
int media=sc/nc;
int ok=0;
while (!ok && x)
{ int cif=x%10;

ok=cif==media;
x/=10;
}
return ok;
}

```

Varianta PASCAL

```

procedure cifre(x:longint;
var nc,sc:integer);

```

```

function subp(x:longint):integer;
var nc,sc,media,cif,ok:integer;
begin
cifre(x,nc,sc);
media:=sc div nc;ok:=0;
while (ok=0) and (x>0) do
begin
cif:=x mod 10;
if cif=media then ok:=1;
x:=x div 10;
end;
subprog:=ok;
end;

```

Varianta <47 intensiv>

SUBIECTUL I

1. d

a) 7

b) 60

c)

```

citeste n (numar natural)
max←0
cat timp n≠0 executa
n←[n/10]
daca max<n%10 atunci
max←n%10
sfarsit daca
sfarsit cat timp
scrie max

```

d)

Varianta C/C++

```
#include <iostream.h>
void main()
{unsigned int n,max;

    cout<<"n=";cin>>n;
    max=0;
    do
    { n/=10 ;
      if (max<n%10)
        max=n%10 ;
    }while(n);
    cout<<max;
}
```

Varianta PASCAL

```
program d2;
var n,max:word;
begin

write('n=');readln(n);
max:=0;
repeat
  n:=n div 10;
  if max<n mod 10 then
    max:=n mod 10;
until n=0;
writeln(max);
end.
```

SUBIECTUL II

1. 6

3. 1

4. C++: (f.a%k==0 && f.b%k==0)

5.

Varianta C/C++

```
#include<iostream.h>
void main()
{unsigned
n,m,i,j,a[11][11],b[11][11];
  cout<<"m=";cin>>m;
  cout<<"n=";cin>>n;
  for (i=1;i<=m;i++)
    for (j=1;j<=n;j++)
      cin>>a[i][j];
  for (i=1;i<=n;i++)
    for (j=1;j<=m;j++)
      b[i][j]=a[j][n-i+1];

  for(i=1;i<=n;i++)
  {cout<<endl;
    for(j=1;j<=m;j++)
      cout<<b[i][j]<<" ";
  }
}
```

Pascal: (f.a mod k=0) and (f.b mod k=0)

Varianta PASCAL

```
program cinci;
var n,m,i,j:byte;
    a,b:array[1..10,1..10] of byte;
begin
write('m=');readln(m);
write('n=');readln(n);
for i:=1 to m do
  for j:=1 to n do
    readln(a[i,j]);
for i:=1 to n do
  for j:=1 to m do
    b[i,j]:=a[j,n-i+1];
for i:=1 to n do
  begin
  writeln;
  for j:=1 to m do write(b[i,j],
  ');
  end;
end.
```

SUBIECTUL III

1. d

2. 168

3.

Varianta C/C++

```
#include<fstream.h>
fstream f("bac.txt",ios::in);
void main()

{unsigned int n,x,y,g,k,nr=0,i;
  cin>>n;
  cin>>k;
  for (i=1;i<=n;i++)
```

Varianta PASCAL

```
program trei;
var n,i,x,y,k,g,nr:integer;
    f:text;

begin
assign(f,'bac.txt');
reset(f);
readln(f,n); readln(k);
```

```

{f>>x ;
 y=x; g=0 ;
 while (y && !g)
   {if (y%10==k) g=1;
    y/=10;}
   if (g) {cout<<x<<" "; nr++;}
 }
 f.close();
 if (!nr) cout<<"NU";
 }

```

```

nr:=0;
for i:=1 to n do
 begin
  read(f,x); y:=x;g:=0;
  while (y<>0) and (g=0) do
   begin
    if y mod 10=k then g:=1;
    y:=y div 10;
   end;
   if g=1 then begin
    write(x,' ');
    inc(nr);
   end;
  end;
 close(f);
 if nr=0 then writeln('NU');
 end.

```

4.

a) C++

```
void cif(long nr, int &s)
```

b)

Varianta C/C++

```

#include<iostream.h>
void main()
{long v[25];
 int w[25],n,i,max=0,s;
 cin>>n;
 for (i=1;i<=n;i++)
  {cin>>v[i];
   s=0;
   cif(v[i],s);
   w[i]=s;
   if (s>max) max=s;
  }
 for (i=1;i<=n;i++)
  if (max==w[i])
   cout<<v[i]<<" ";
 }

```

PASCAL procedure cif(nr:longint;var s:byte);

Varianta PASCAL

```

program patru;
var n,i,max,s:byte;
    v:array[1..25] of longint;
    w:array[1..25] of byte;
begin
 readln(n);
 max:=0;
 for i:=1 to n do
  begin
   read(v[i]);
   s:=0;
   cif(v[i],s);
   w[i]:=s;
   if s>max then max:=s;
  end;
 for i:=1 to n do
  if max=w[i] then
   write(v[i],' ');
 end.

```

SUBIECTUL I

1. d

2.

a) 7

b) 60

c)

```

citeste n (numar natural)
max←0
cat timp n≠0 executa
  n←[n/10]
  daca max<n%10 atunci
    max←n%10
sfarsit daca

```

Varianta <47 neintensiv>

sfarsit cat timp
scrie max

d)

Varianta C/C++

```
#include <iostream.h>

void main()
{ unsigned int n,max;
  cout<<"n=";cin>>n;
  max=0;
  do
  { n/=10 ;
    if (max<n%10)
      max=n%10 ;
  }while(n);
  cout<<max;
}
```

Varianta PASCAL

```
program d2;
var n,max:word;
begin
  write('n=');readln(n);
  max:=0;
  repeat
    n:=n div 10;
    if max<n mod 10 then
      max:=n mod 10;
  until n=0;
  writeln(max);
end.
```

SUBIECTUL II

1. c
2. b
3. acalaureat
4. C++: (f.a%k==0 && f.b%k==0)
- 5.

Varianta C/C++

```
#include<iostream.h>
void main()
{ unsigned int
n,m,i,j,k,a[11][11],nr;
  cout<<"m=";cin>>m;
  cout<<"n=";cin>>n;
  for (i=1;i<=m;i++)
    for (j=1;j<=n;j++)
      cin>>a[i][j];
  nr=0;
  for (i=m;i>=1;i--)
    if (i%2)
      {for (k=i;k<=m-nr-1;k++)
        for (j=1;j<=n;j++)
          a[k][j]=a[k+1][j];
        nr++;}
  m=nr;
  for(i=1;i<=m;i++)
    {cout<<endl;
     for(j=1;j<=n;j++)
       cout<<a[i][j]<<" ";
    }
}
```

Pascal: (f.a mod k=0) and (f.b mod k=0)

Varianta PASCAL

```
program cinci;
var n,m,i,j,k,nr:byte;
    a:array[1..10,1..10] of byte;
begin
  write('m=');readln(m);
  write('n=');readln(n);
  for i:=1 to m do
    for j:=1 to n do
      readln(a[i,j]);
  nr:=0;
  for i:=m downto 1 do
    if i mod 2=1 then
      begin
        for k:=i to m-nr-2 do
          for j:=1 to n do
            a[k,j]:=a[k+1,j];
          inc(nr);
        end;
      dec(m,nr);
      for i:=1 to m do
        begin
          writeln;
          for j:=1 to n do write(a[i,j],' ');
        end;
      end.
```

SUBIECTUL III

1. d
2. 168

3.

Varianta C/C++

```
#include<fstream.h>
fstream f("bac.txt",ios::in);
void main()
{
    unsigned int n,x,y,g,k,nr=0,i;
    f>>n;
    cin>>k;
    for (i=1;i<=n;i++)
        {f>>x ;
         y=x; g=0 ;
         while (y && !g)
             {if (y%10==k) g=1;
              y/=10;}
         if (g) {cout<<x<<" "; nr++;}
        }
    f.close();
    if (!nr) cout<<"NU";
}
```

Varianta PASCAL

```
program trei;
var n,i,x,y,k,g,nr:integer;
    f:text;

begin
    assign(f,'bac.txt');
    reset(f);
    readln(f,n); readln(k);
    nr:=0;
    for i:=1 to n do
        begin
            read(f,x); y:=x;g:=0;
            while (y<>0) and (g=0) do
                begin
                    if y mod 10=k then g:=1;
                    y:=y div 10;
                end;
            if g=1 then begin
                write(x,' ');
                inc(nr);
            end;
        end;
    close(f);
    if nr=0 then writeln('NU');
end.
```

4.

a) C++ void cif(long nr, int &s)

b)

Varianta C/C++

```
#include<iostream.h>
void main()
{long v[25];
 int w[25],n,i,max=0,s;
 cin>>n;
 for (i=1;i<=n;i++)
 {cin>>v[i];
  s=0;
  cif(v[i],s);
  w[i]=s;
  if (s>max) max=s;
 }

 for (i=1;i<=n;i++)
 if (max==w[i])
 cout<<v[i]<<" ";
}
```

PASCAL procedure cif(nr:longint;var s:byte);

Varianta PASCAL

```
program patru;
var n,i,max,s:byte;
    v:array[1..25] of longint;
    w:array[1..25] of byte;

begin
    readln(n);
    max:=0;
    for i:=1 to n do
        begin
            read(v[i]);
            s:=0;
            cif(v[i],s);

            w[i]:=s;
            if s>max then max:=s;
        end;
    for i:=1 to n do
        if max=w[i] then
            write(v[i],' ');
    end.
```

Varianta <48 intensiv>

SUBIECTUL I

1. a

2. a) 8, 905, 707, 801, 10001, 105

b) 101, 703, 209, 904

c)

```

citeste n (numar natural nenul)
i←1
repetă
  citeste x (numar natural)
  nr←0
  cat timp x>0 executa
    nr←nr*100+x%10
    x←[x/100]
  sfarsit cat timp
  cat timp nr>0 executa
    x←x*10+nr%10
    nr←[nr/10]
  sfarsit cat timp
  scrie x
  i←i+1
pana cand i>n
    
```

d)

Varianta C/C++

```

#include <iostream.h>
void main()
{unsigned int n,x,nr;
 cout<<"n=";<<cin>>n;
 for (i=1;i<=n;i++)
  {cin>>x;
   nr=0;
   while (x>0)
    {nr=nr*100+x%10;
     x/=100;
    }

   while (nr>0)
    {x=x*10+nr%10;
     nr/=10;
    }
   cout<<x;
  }
}
    
```

Varianta PASCAL

```

program d2;
var n,x,nr:word;
begin
 write('n=');readln(n);
 for i:=1 to n do
  begin
   readln(x);
   nr:=0;
   while x>0 do
    begin
     nr:=nr*100+x%10;
     x:=x div 100;
    end;
   while nr>0 do
    begin
     x:=x*10+nr%10;
     nr:=nr div 10;
    end;
   write(x,' ');
  end;
end.
    
```

SUBIECTUL II

1. d

3. 3

4. $\sqrt{a.x^2+a.y^2}$

5.

Varianta C/C++

```

#include<iostream.h>
void main()
{unsigned int n,i,j,a[10][10];
 cout<<"n=";<<cin>>n;
 for (i=1;i<=n;i++)
  for (j=1;j<=n;j++)
   a[i][j]=2*((i-1)*n+j);
 for(i=1;i<=n;i++)
  {cout<<endl;
    
```

Varianta PASCAL

```

program cinci;
var n,i,j:byte;
    a:array[1..10,1..10] of byte;
begin
 write('n=');readln(n);
 for i:=1 to n do
  for j:=1 to n do
   a[i,j]:=2*((i-1)*n+j);
 for i:=1 to n do
    
```

```

    for(j=1;j<=n;j++)
        cout<<a[i][j]<<" ";
}
}

```

```

begin
writeln;
for j:=1 to n do write(a[i,j], ' ');
end;
end.

```

SUBIECTUL III

1. c

2. M1, M3, M2, M4

3.

a)

Varianta C/C++

```

int div(unsigned int x,unsigned int
y)
{while (x!=y)
    if (x>y) x=x-y;
    else y=y-x;
return x ; }

```

Varianta PASCAL

```

function div1(x,y :byte) :byte ;
begin
    while x<>y do
        if x>y then x:=x-y;
        else y:=y-x;

    div1:=x;
end;

```

b)

Varianta C/C++

```

#include<iostream.h>
void main()
{unsigned int n,a,b,i,x=0,v[100];
cin>>a>>b>>n;
if (a>b) {a=a+b;b=a-b;a=a-b;}
for (i=a;i<=b;i++)
    if (div(i,n)==1) v[++x]=i;
for (i=1;i<=x;i++)
    cout<<v[i]<<" ";
}

```

Varianta PASCAL

```

program trei;
var n,i,a,b,x:integer;
    v:array[1..100] of byte;
begin
readln(a,b,n);
if a>b then
    begin
        a:=a+b;
        b:=a-b;
        a:=a-b;
    end;
for i:=a to b do
    if div1(i,n)=1 then begin
        inc(x);
        v[x]:=i;
    end;
for i:=1 to x do
    write(v[i], ' ');
end.

```

4.

Varianta C/C++

```

#include<fstream.h>
void main()
{unsigned int v[5000],n,i,x,a,j,nr;
fstream f("bac.in",ios::in);
fstream g("bac.out",ios::out);
f>>n;
for (i=1;i<=n;i++)
    f>>v[i];
do
{x=1;
for (i=1;i<=n-1;i++)
    if (v[i]>v[i+1])
        {a=v[i];v[i]=v[i+1];v[i+1]=a;
        x=0;}
}while (x==1);
}

```

Varianta PASCAL

```

program patru;
var n,i,a,j,nr:word;
    x:Boolean;
    v:array[1..5000] of word;
    f,g:text;
begin
assign(f,'bac.in'); reset(f);
assign(g,'bac.out'); rewrite(g);
readln(f,n);
for i:=1 to n do
    read(f,v[i]);
repeat
    x:=true;
    for i:=1 to n-1 do
        if v[i]>v[i+1] then

```

```

i=1;
while (i<=n)
  {nr=1;
  j=i+1;
  while (v[j]==v[i]&& j<=n)
    {j++;nr++;}
  if (nr==1) g<<v[i]<<" ";
  i=j;
  }
f.close();
g.close();
}

```

```

begin
a:=v[i];v[i]:=v[i+1];v[i+1]:=a;
  x:=false;
  end;
until x=true;
i:=1;
while i<=n do
  begin
  nr:=1; j:=i+1;
  while (j<=n) and (v[j]=v[i]) do
    begin
    inc(j);inc(nr);
    end;
  if nr=1 then write(g,v[i],' ');
  end;
close(f); close(g);
end.

```

Varianta <48 neintensiv>

SUBIECTUL I

1. a

2.

a) 8, 905, 707, 801, 10001, 105

b) 101, 703, 209, 904

c)

citeste n (numar natural nenul)

i←1

repetă

citeste x (numar natural)

nr←0

cat timp x>0 executa

nr←nr*100+x%10

x←[x/100]

sfarsit cat timp

cat timp nr>0 executa

x←x*10+nr%10

nr←[nr/10]

sfarsit cat timp

scrie x

i←i+1

pana cand i>n

d)

Varianta C/C++

```
#include <iostream.h>
```

```
void main()
```

```
{unsigned int n,x,nr;
```

```
cout<<"n=";cin>>n;
```

```
for (i=1;i<=n;i++)
```

```
{cin>>x;
```

```
nr=0;
```

```
while (x>0)
```

```
{nr=nr*100+x%10;
```

```
  }x/=100;
```

```
while (nr>0)
```

```
{x=x*10+nr%10;
```

Varianta PASCAL

```
program d2;
```

```
var n,x,nr:word;
```

```
begin
```

```
write('n=');readln(n);
```

```
for i:=1 to n do
```

```
begin
```

```
readln(x);
```

```
nr:=0;
```

```
while x>0 do
```

```
begin
```

```
nr:=nr*100+x%10;
```

```
x:=x div 100;
```

```
end;
```

```

        nr/=10;
    }
    cout<<x;
}
}

```

```

while nr>0 do
begin
    x:=x*10+nr%10;
    nr:=nr div 10;
end;
write(x,' ');
end;
end.

```

SUBIECTUL II

1. d

2. a

3. 3

4. $\sqrt{a.x*a.x+a.y*a.y}$

5.

Varianta C/C++

```

#include<iostream.h>
void main()
{unsigned int n,i,j,a[10][10];
  cout<<"n=";cin>>n;
  for (i=1;i<=n;i++)
    for (j=1;j<=n;j++)
      a[i][j]=2*((i-1)*n+j);
  for(i=1;i<=n;i++)
    {cout<<endl;
     for(j=1;j<=n;j++)
       cout<<a[i][j]<<" ";
    }
}

```

Varianta PASCAL

```

program cinci;
var n,i,j:byte;
    a:array[1..10,1..10] of byte;
begin
write('n=');readln(n);
for i:=1 to n do
  for j:=1 to n do
    a[i,j]:=2*((i-1)*n+j);
for i:=1 to n do
  begin
  writeln;
  for j:=1 to n do write(a[i,j],' ');
  end;
end.

```

SUBIECTUL III

1. c

2. M1, M3, M2, M4

3.

a)

Varianta C/C++

```

int div(unsigned int x,unsigned int
y)
{while (x!=y)
  if (x>y) x=x-y;
  else y=y-x;
return x ; }

```

Varianta PASCAL

```

function div1(x,y :byte) :byte ;
begin
  while x<>y do
    if x>y then x:=x-y;
    else y:=y-x;
  div1:=x;
end;

```

b)

Varianta C/C++

```

#include<iostream.h>
void main()
{unsigned int n,a,b,i,x=0,v[100];
  cin>>a>>b>>n;
  if (a>b) {a=a+b;b=a-b;a=a-b;}
  for (i=a;i<=b;i++)
    if (div(i,n)==1) v[++x]=i;
  for (i=1;i<=x;i++)
    cout<<v[i]<<" ";
}

```

Varianta PASCAL

```

program trei;
var n,i,a,b,x:integer;
    v:array[1..100] of byte;
begin
readln(a,b,n);
if a>b then
  begin
  a:=a+b;
  b:=a-b;
  a:=a-b;

```

4.

Varianta C/C++

```
#include<fstream.h>
void main()
{unsigned int v[5000],n,i,x,a,j,nr;
 fstream f("bac.in",ios::in);
 fstream g("bac.out",ios::out);
 f>>n;
 for (i=1;i<=n;i++)
  f>>v[i];
 do
 {x=1;
  for (i=1;i<=n-1;i++)
   if (v[i]>v[i+1])
    {a=v[i];v[i]=v[i+1];v[i+1]=a;
     x=0;}
 }while (x==1);
 i=1;
 while (i<=n)
  {nr=1;
   j=i+1;
   while (v[j]==v[i]&& j<=n)
    {j++;nr++;}
   if (nr==1) g<<v[i]<<" ";
   i=j;
  }
 f.close();
 g.close();
 }
```

SUBIECTUL I

1. b

2.

a) 204

b) 90807

c)

citeste x (numar natural nenul)

k_0
~~Pe~~eta

$k \leftarrow k \cdot 10 + x \% 10$

$x \leftarrow [x / 10]$

```
end;
for i:=a to b do
  if div1(i,n)=1 then begin
    inc(x);
    v[x]:=i;
  end;

for i:=1 to x do
  write(v[i],' ');
end.
```

Varianta PASCAL

```
program patru;
var n,i,a,j,nr:word;
    x:Boolean;
    v:array[1..5000] of word;
    f,g:text;
begin
  assign(f,'bac.in'); reset(f);
  assign(g,'bac.out'); rewrite(g);
  readln(f,n);
  for i:=1 to n do
    read(f,v[i]);
  repeat
    x:=true;
    for i:=1 to n-1 do
      if v[i]>v[i+1] then
        begin
          a:=v[i];v[i]:=v[i+1];v[i+1]:=a;
          x:=false;
        end;
  until x=true;
  i:=1;
  while i<=n do
    begin
      nr:=1; j:=i+1;
      while (j<=n) and (v[j]=v[i]) do
        begin
          inc(j);inc(nr);
        end;
      if nr=1 then write(g,v[i],' ');
    end;
  close(f); close(g);
end.
```

Varianta <49 intensiv>

```
pana cand x=0
repete
  x←x*10+k%10
  k←[k/100]
pana cand k=0
scrie x
```

d)

Varianta C/C++

```
#include <iostream.h>
void main()
{unsigned int x,k;
  cout<<"x=";cin>>x;
  k=0;
  while (x)
  {k=k*10+x%10;
   x=x/10;
  }
  while (k)
  {x=x*10+k%10;
   k=k/100;
  }
  cout<<x;
}
```

Varianta PASCAL

```
program d2;
var x,k:word;
begin
  write('x=');readln(x);
  k:=0;
  while x<>0 do
  begin
    k:=k*10+x mod 10;
    x:=x div 10;
  end;
  while k<>0 do
  begin
    x:=x*10+k mod 10;
    k:=k div 100;
  end;
  writeln(x);
end.
```

SUBIECTUL II

1. d

2. c

3. $(x.med1+x.med2)/2$

5.

Varianta C/C++

```
#include<iostream.h>
#include<string.h>
void main()
{char s[21],t[21],p[]="aeiou";
  int i;
  cin.get(s,21);
  strcpy(t,"");
  for (i=0;i<strlen(s);i++)
    if (strchr(p,s[i]))
      strncat(t,s+i,i);
  cout<<t;
}
```

Varianta PASCAL

```
program cincii;
var s,t:string[20];
begin
  readln(s);
  t:='';
  for i:=1 to length(s) do
    if (s[i]='a') or (s[i]='e') or
      (s[i]='i') or (s[i]='o') or
      (s[i]='u') then
      t:=t+s[i];
  write(t);
end.
```

SUBIECTUL III

1. c

2. 4

3.

Varianta C/C++

```
#include<iostream.h>
void main()
{unsigned int a,b,n,i,x=0,v[100];
  cin>>a>>b>>n;
```

Varianta PASCAL

```
program trei;
var a,b,h,i,x:word;
    v:array[1..100] of word;
begin
```

```

if (a>b) {a=a+b;b=a-b;a=a-b;}
while (a%n) a++;
i=a/n;
while (i*n<=b)
  {v[++x]=i*n;
  i++;
}
if (!x) cout<<"NU";
else
  for (i=1;i<=x;i++)
    cout<<v[i]<<" ";
}

```

4.

a)

Varianta C/C++

void cmax(unsigned int a, unsigned int &b)

b)

Varianta C/C++

```

#include<fstream.h>
fstream f("bac.txt",ios::in);
void cmax(unsigned int a, unsigned int &b)
{unsigned int max=0;
while (a)
  {if (a%10>max) max=a%10;
  a/=10;
}
b=max;
}

void main()
{unsigned int x,p=0,b;;
while (f>>x)
  {cmax(x,b);
  if (b>p) p=b;
}
cout<<p;
}

```

```

readln(a,b,n);
x:=0;
if a>b then
  begin
    a:=a+b;b:=a-b;a:=a-b;
  end;
while a mod n<>0 do inc(a);
i:=a div n;
while i*n<=b do
  begin
    if i*n>=a then
      begin
        inc(x);
        v[x]:=i*n;
      end;
    inc(i);
  end;
if x=0 then writeln('NU')
else
  for i:=1 to x do
    write(v[i],' ');
end.

```

Varianta PASCAL

procedure cmax(a:word; var b:word);

Varianta PASCAL

```

program patru;
var x,p,b:word;
f:text;
procedure cmax(a:word; var b:word);
var max:word;
begin
  max:=0;
  while a<>0 do
    begin
      if a mod 10>max then max:=a mod 10;
      a:=a div 10;
    end;
  b:=max;
end;
begin
  assign(f,'bac.txt');
  reset(f);
  p:=0;
  while not(eof(f)) do
    begin
      read(f,x);
      cmax(x,b);
      if b>p then p:=b;
    end;
  write(p);
end.

```

Varianta <49 neintensiv>

SUBIECTUL I

1. b

2.

b) 20807

c)

citeste x (numar natural nenul)

k←0

repete

 k←k*10+x%10

 x←[x/10]

pana cand x=0

repete

 x←x*10+k%10

 k←[k/100]

pana cand k=0

scrie x

d)

Varianta C/C++

```
#include <iostream.h>
```

```
void main()
```

```
{unsigned int x,k;
```

```
  cout<<"x=";cin>>x;
```

```
  k=0;
```

```
  while (x)
```

```
    {k=k*10+x%10;
```

```
    x=x/10;
```

```
  }
```

```
while (k)
```

```
  {x=x*10+k%10;
```

```
  k=k/100;
```

```
  }
```

```
  cout<<x;
```

```
}
```

SUBIECTUL II

1. d

2. c

3. $(x.med1+x.med2)/2$

5.

Varianta C/C++

```
#include<iostream.h>
```

```
#include<string.h>
```

```
void main()
```

```
{char s[21],t[21],p[]="aeiou";
```

```
  int i;
```

```
  cin.get(s,21);
```

```
  strcpy(t,"");
```

```
  for (i=0;i<strlen(s);i++)
```

```
    if (strchr(p,s[i]))
```

```
      strncat(t,s+i,1);
```

```
  cout<<t;
```

```
}
```

168

Varianta PASCAL

```
program d2;
```

```
var x,k:word;
```

```
begin
```

```
  write('x=');readln(x);
```

```
  k:=0;
```

```
  while x<>0 do
```

```
    begin
```

```
      k:=k*10+x mod 10;
```

```
      x:=x div 10;
```

```
    end;
```

```
  while k<>0 do
```

```
    begin
```

```
      x:=x*10+k mod 10;
```

```
      k:=k div 100;
```

```
    end;
```

```
  writeln(x);
```

```
end.
```

Varianta PASCAL

```
program cinci;
```

```
var s,t:string[20];
```

```
begin
```

```
  readln(s);
```

```
  t:='';
```

```
  for i:=1 to length(s) do
```

```
    if (s[i]='a') or (s[i]='e') or
```

```
      (s[i]='i') or (s[i]='o') or
```

```
      (s[i]='u') then
```

```
      t:=t+s[i];
```

```
  write(t);
```

```
end.
```


SUBIECTUL III

1. c

2. 4

3.

Varianta C/C++

```
#include<iostream.h>
void main()
{unsigned int a,b,n,i,x=0,v[100];
 cin>>a>>b>>n;
 if (a>b) {a=a+b;b=a-b;a=a-b;}
 while (a%n) a++;
 i=a/n;
 while (i*n<=b)
 {v[++x]=i*n;
 i++;
 }
 if (!x) cout<<"NU";

 else
 for (i=1;i<=x;i++)
 cout<<v[i]<<" ";
 }
```

4.

a)

Varianta C/C++

void cmax(unsigned int a, unsigned int &b)

b)

Varianta C/C++

```
#include<fstream.h>
fstream f("bac.txt",ios::in);
void cmax(unsigned int a, unsigned
int &b)
{unsigned int max=0;
 while (a)
 {if (a%10>max) max=a%10;
 a/=10;
 }
 b=max;
 }

void main()
{unsigned int x,p=0,b;;
 while (f>>x)
 {cmax(x,b);
 if (b>p) p=b;
 }
 cout<<p;
 }
```

Varianta PASCAL

```
program trei;
var a,b,n,i,x:word;
 v:array[1..100] of word;
begin
 readln(a,b,n);
 x:=0;
 if a>b then
 begin
 a:=a+b;b:=a-b;a:=a-b;
 end;
 while a mod n<>0 do inc(a);
 i:=a div n;

 while i*n<=b do
 begin
 if i*n>=a then
 begin
 inc(x);
 v[x]:=i*n;
 end;
 inc(i);
 end;
 if x=0 then writeln('NU')
 else
 for i:=1 to x do
 write(v[i],' ');
 end.
```

Varianta PASCAL

procedure cmax(a:word; var b:word);

Varianta PASCAL

```
program patru;
var x,p,b:word;
 f:text;
procedure cmax(a:word; var b:word);
var max:word;
begin
 max:=0;
 while a<>0 do
 begin
 if a mod 10>max then max:=a mod
10;
 a:=a div 10;
 end;
 b:=max;
 end;
begin
 assign(f,'bac.txt');
 reset(f);
 p:=0;
 while not(eof(f)) do
```

```

begin
  read(f, x);
  cmax(x, b);
  if b>p then p:=b;
end;
write(p);
end.

```

Varianta <50>

SUBIECTUL I

1. b

2. a) 2

b) Se poate citi orice set de patru valori care conține la cifra zecilor cifra 9. Exemplu: 19, 192, 396, 999

c) $i \leftarrow 1$

executa

```

citeste x
c ← [x/10] % 10
daca c < k atunci
    k ← c

```

$i \leftarrow i + 1$

cat timp $i \leq n$

scrie k

d) Varianta C/C++

```

#include<iostream.h>
void main()
{
  unsigned int n, k, x, c;
  cout<<"n="; cin>>n;
  k=9;
  for(int i=1; i<=n; i++)
  {
    cout<<"x="; cin>>x;
    c=(x/10)%10;
    if(c<k)
      k=c;
  }
  cout<<k;
}

```

d) Varianta PASCAL

```

program exp;
var n, k, c, i, x: integer;
begin
  write('n='); readln(n);
  k:=9;
  for i:=1 to n do
    begin
      write('x='); readln(x);
      c:=(x div 10) mod 10;
      if c<k then
        k:=c;
    end;
  writeln(k);
end.

```

SUBIECTUL II

1. apentru intensiv și neintensiv

2. dpentru neintensiv și intensiv

3.6 noduri, adică ăcelele noduri care sunt și frunze (9, 10, 6, 7, 8, 4)

4. Varianta C/C++

Intensiv: Ultima cifră este 5. Orice număr impar înmulțit cu 5, va avea ultima cifră 5.

Neintensiv: Variabila are reține

caracterul 'r'

5. Varianta C/C++

Se caută în șir prima apariție a unei vocale

Varianta PASCAL

Intensiv: Ultima cifră este 5. Orice număr impar înmulțit cu 5, va avea ultima cifră 5.

Neintensiv: Variabila are reține

caracterul 'r'

Varianta PASCAL

Se caută în șir prima apariție a unei vocale

după care se elimină folosind un șir auxiliar m. Algoritmul se repetă pentru ultima apariție a unei vocale parcurgând șirul de la sfârșit la început.

```
#include<iostream.h>
#include<string.h>
void main()
{
char s[21],*p, voc[]="aeiou",m[21];
cout<<"dati sirul ";cin>>s;
int ok=1;
for(int i=0;i<strlen(s)&&ok;i++)
    if(strchr(voc,s[i]))
        ok=0;
m[0]=0;
strncat(m,s,i-1);

strcat(m,s+i);
strcpy(s,m);
ok=1;
for(i=strlen(s)-1;i>=0&&ok;i--)
    if(strchr(voc,s[i]))
        ok=0;
m[0]=0;
strncat(m,s,i+1);
strcat(m,s+i+2);
strcpy(s,m);
cout<<s;
}
```

după care se elimină folosind un șir auxiliar m. Algoritmul se repetă pentru ultima apariție a unei vocale parcurgând șirul de la sfârșit la început.

```
program bacsir;
type litere=set of 'a'..'z';
var voc:litere;
    s:string[20];
    i:byte;
    ok:boolean;
begin
    voc=['a','e','i','o','u'];
    write('s=');readln(s);
    i:=1;
    ok:=true;
    while (i<=length(s)) and
(ok=true) do
        begin
            if s[i] in voc then
                begin
                    delete(s,i,1);
                    ok:=false;
                end;
            i:=i+1;
        end;
    i:=length(s);
    ok:=true;
    while (i>=1) and (ok=true) do
        begin
            if s[i] in voc then
                begin
                    delete(s,i,1);
                    ok:=false;
                end;
            i:=i-1;
        end;
    write(s);
end.
```

SUBIECTUL III

1. b pentru intensiv și neintensiv

2. 332321

3. Varianta C/C++

3.a unsigned int div(unsigned int x, unsigned int y)

```
{
    if(x%y==0)
        return 1;
    return 0;
}
```

Varianta PASCAL

function div1(x,y:integer):boolean;

```
begin
    if x mod y=0 then
        div1:=true
    else
        div1:=false;
end;
```

3.b. Se citesc cu validare toate cele trei numere pentru a îndeplini condiția ca ele să fie formate din maxim trei cifre. Dacă a fost citit valoarea lui $a > b$, se va efectua interschimbarea celor două valori. Pentru intervalul închis $[a,b]$ se parcurge cu un for

mulțimea numerelor, iar cele care sunt divizori ale lui n vor fi introduse în vectorul v pe poziția k .

```
#include<iostream.h>
unsigned int div(unsigned int x,
unsigned int y)

{if(x%y==0)
    return 1;
    return 0;
}
void main()
{
    unsigned int v[500],a,b,n,k,i;
    do
    {
        cout<<"n=";cin>>n;
        cout<<"a=";cin>>a;
        cout<<"b=";cin>>b;
    }
    while(a>999||b>999||n>999);
    if(b<a)
    {
        unsigned int aux=a;
        a=b;
        b=aux;
    }
    k=0;
    for(i=a;i<=b;i++)
        if(div(n,i))
            v[++k]=i;
    for(i=1;i<=k;i++)
        cout<<v[i]<<" ";
}
```

```
program bac;
var v:array[1..500] of word;
    i,n,aux,a,b,k:word;

function div1(x,y:word):boolean;
begin
    if x mod y=0 then
        div1:=true
    else
        div1:=false;
    end;
begin
    repeat
        write('n=');readln(n);
        write('a=');readln(a);
        write('b=');readln(b);

        until (a<=999) or (b<=999) or
(n<=999);
        if b<a then
            begin
                aux:=a;
                a:=b;
                b:=aux;
            end;
        k:=0;
        for i:=a to b do
            if div1(n,i)=true then
                begin
                    k:=k+1;
                    v[k]:=i;
                end;
        for i:=1 to k do
            write(v[i],' ');
    end.
```

4) Varianta C/C++

Se utilizează un vector de 10000 de elemente care se initializează cu 0. Odată citit un număr x se semnalează apariția un vectorului caracteristic. Pentru elementele care au fost citite de mai multe ori valoarea din vector ≥ 2 . Doar acele valori vor fi afișate în fișier.

```
#include<fstream.h>
unsigned int n,v[10000],x;
void main()
{
    for(int i=0;i<10000;i++)
        v[i]=0;
    ifstream f("bac.in");
    ofstream g("bac.out");
    f>>n;
    for(i=1;i<=n;i++)
    {
        f>>x;
        v[x]++;
    }
    for(i=0;i<10000;i++)
        if(v[i]>=2)
            g<<i<<" ";
}
```

Varianta PASCAL

```
program bac2;
var n,x,i:integer;
    v:array[1..10000] of integer;
    f,g:text;
begin
    for i:=1 to 10000 do
        v[i]:=0;
    assign(f,'bac.in');reset(f);
    assign(g,'bac.out');rewrite(g);
    readln(f,n);
    for i:=1 to n do
        begin
            read(f,x);
            v[x]:=v[x]+1;
        end;
    for i:=1 to 10000 do
        if v[i]>=2 then
```

```
f.close();
g.close();
}
```

```
write(g,i,' ');
close(g);
close(f);
end.
```

Varianta <51>

SUBIECTUL I

1. d

2. a) 4220

b) Orice număr format numai din cifre pare cu ultima cifră diferită de 0: 2468

c)

```
citeste x
      (numar natural nenul)
```

```
z ← 0
```

```
cat timp x ≠ 0 executa
```

```
  c ← x % 10
```

```
  daca c % 2 ≠ 0 atunci
```

```
    z ← z * 10 + c - 1
```

```
  altfel
```

```
    z ← z * 10 + c
```

```
  sfarsit daca
```

```
  x ← [x / 10]
```

```
sfarsit cat timp
```

```
scrie z
```

d) Varianta C/C++

```
#include<iostream.h>
```

```
unsigned int x,z,c;
```

```
void main()
```

```
{
```

```
  cin >> x;
```

```
  do{
```

```
    c = x % 10;
```

```
    if(c % 2 == 1)
```

```
      z = z * 10 + c - 1;
```

```
    else
```

```
      z = z * 10 + c;
```

```
    x = x / 10;
```

```
  }while(x);
```

```
  cout << z; }
```

Varianta PASCAL

```
program bac;
```

```
var x,z:word;
```

```
    c:byte;
```

```
begin
```

```
  readln(x);
```

```
  repeat
```

```
    c := x mod 10;
```

```
    if c mod 2 = 1 then
```

```
      z := z * 10 + c - 1
```

```
    else
```

```
      z := z * 10 + c;
```

```
    x := x div 10;
```

```
  until x = 0;
```

```
end; writeln(z);
```

SUBIECTUL II

1. a **pentru C/C++**

2. d

3.12

4. **Pentru intensiv**

6 muchii

5. **Pentru intensiv**

Se construiește șirul **s1** astfel: parcurgând șirul **s** începând cu primul caracter, astfel încât fiecare

b **pentru Pascal**

Pentru neintensiv

Rădăcina: nodul 2

Noduri terminale: 1, 4, 5, 8, 10

Pentru neintensiv

Presupunem că șirul este format numai din litere și spații. Parcurgem șirul și

caracter din sirul `s` din pozitia `i` se adauga de `i+1` ori la sirul `s1`.

Varianta C/C++

Program intensiv

```
#include<iostream.h>
#include<string.h>
char s[53],s1[255];
void main()
{
cout<<"dati sirul";
cin.get(s,53);
for(int i=0;i<strlen(s)-1; i=i+2)
    for(int j=1;j<=s[i+1]-48;j++)
        strcat(s1,s+i,1);
cout<<s1<<endl;
}
```

pentru fiecare caracter verificăm dacă este contrazisă presupunerea inițială.

Varianta PASCAL

Program intensiv

```
program bac;
var s:string[52];
    s1:string;
    i,j:byte;
begin
write('dati sirul');
readln(s);
i:=1;
while(i<length(s)) do
begin
for j:=1 to ord(s[i+1])-48 do
s1:=s1+s[i];
i:=i+2;
end;
writeLn(s1);
end.
```

Varianta C/C++

Pentru neintensiv

```
#include<iostream.h>
#include<ctype.h>
#include<string.h>
char s[31];
int k;
void main()
{
cout<<"dati sirul";
cin.get(s,31);
for(int i=0;i<strlen(s);i++)
if(!isalpha(s[i])&&s[i]!=' ')
k=1;
if(k)
cout<<"NU";
else
cout<<"DA";
}
```

Varianta PASCAL

Pentru neintensiv

```
program bac;
type mult=set of 'A'..'z';
var s:string[30];
    m:mult;
    i:byte;
    k:boolean;
begin
readln(s);
k:=true;
m=['A'..'Z','a'..'z'];
for i:=1 to length(s) do
if (not(s[i] in m)) and
(s[i]<>' ') then
k:=false;
if k=false then
writeln('NU')
else
writeln('DA');
end.
```

SUBIECTUL III

1. a

2. Numarul de permutări pentru un număr de 8 cifre este 8!. Din acesta se scade numărul de configurații cu cifra 0 pe prima poziție, adică 7!. Se obțin 8!-7! numere cu exact 8 cifre.

8!-7!=35280

3. pentru intensiv:

Calculăm distanțele între punctele citite prin apeluri ale funcției `dist2` și le introducem în

vectorul `a`. Dacă punctele citite reprezintă vârfurile unui pătrat, atunci în vector ar trebui să avem 4 valori egale cu minimul din

pentru neintensiv:

Inițializăm variabila locală `d` cu valoarea 2. Cât timp `d` nu divide numărul `a` creștem valoarea lui

`d`. Astfel la ieșirea din structura repetitivă în variabila `d` se va reține primul număr prim care divide pe `a`.

vector, corespunzătoare laturilor și 2 valori corespunzătoare celor doua diagonale, egale cu maximul din vector.

Varianta C/C++

a) Pentru intensiv

```
unsigned long dist2(int xa,int ya,int
xb,int yb)
{return pow(xb-xa,2)+pow(yb-ya,2);
}
```

b)

```
#include<iostream.h>
#include<math.h>
int xa,xb,xc,xd,ya,yb,yc,yd;
unsigned long a[6],min,max,k1=1,k2=1;
unsigned long dist2(int xa,int ya,int
xb,int yb)
{return pow(xb-xa,2)+pow(yb-ya,2);
}
void main()
{cin>>xa>>ya>>xb>>yb>>xc>>yc>>xd>>yd;
a[0]=dist2(xa,ya,xb,yb);
a[1]=dist2(xa,ya,xc,yc);
a[2]=dist2(xa,ya,xd,yd);
a[3]=dist2(xb,yb,xc,yc);
a[4]=dist2(xb,yb,xd,yd);
a[5]=dist2(xc,yc,xd,yd);
min=a[0];
max=a[0];
for(int i=1;i<=5;i++)
{if(min>a[i])
{min=a[i];
k1=1;
}
else
if(min==a[i])
k1++;
if(max<a[i])
{max=a[i];
k2=1;
}
else
if(max==a[i])
k2++;
}
if(k1!=4||k2!=2)
cout<<"NU";
else
cout<<"DA";
}
```

Varianta PASCAL

a) Pentru intensiv

```
function
dist2(xa,ya,xb,yb:integer):longint;
begin
dist2:=sqr(xb-xa)+sqr(yb-ya);
end;
```

b)

```
program bac;
var
xa,ya,xb,yb,xc,yc,xd,yd,k1,k2,i:integer;
min,max:longint;
a:array[1..6] of longint;
function
dist2(xa,ya,xb,yb:integer):longint;
begin
dist2:=sqr(xb-xa)+sqr(yb-ya);
end;
begin
readln(xa); readln(ya); readln(xb);
readln(yb); readln(xc); readln(yc);
readln(xd); readln(yd);
a[1]:=dist2(xa,ya,xb,yb);
a[2]:=dist2(xa,ya,xc,yc);
a[3]:=dist2(xa,ya,xd,yd);
a[4]:=dist2(xb,yb,xc,yc);
a[5]:=dist2(xb,yb,xd,yd);
a[6]:=dist2(xc,yc,xd,yd);
min:=a[1];k1:=1;
max:=a[1]; k2:=1;
for i:=2 to 6 do
begin
if min>a[i] then
begin
min:=a[i];
k1:=1;
end
else
if min=a[i] then
k1:=k1+1;
if max<a[i] then
begin
max:=a[i];
k2:=1;
end
else
if max=a[i] then
k2:=k2+1;
end;
if (k1<>4) or (k2<>2) then
writeln('NU')
else
writeln('DA');
```

	end.
<p>Pentru neintensiv:</p> <p>a)</p> <pre>unsigned int divizor(unsigned int k) {unsigned int d=2; while(k%d) d++; return d; }</pre>	<p>Pentru neintensiv:</p> <p>a)</p> <pre>function divizor(k:word):word; var d:word; begin d:=2; while(k mod d<>0) do d:=d+1; divizor:=d; end;</pre>
<p>b)</p> <pre>#include<fstream.h> unsigned int k,n,s; unsigned int divizor(unsigned int a) {unsigned int d=2; while(a%d) d++; return d; } void main() { ifstream f("date.in"); f>>n; for(int i=1;i<=n;i++) { f>>k; s+=divizor(k); } f.close(); cout<<s; }</pre>	<p>b)</p> <pre>program bac; var k,s:word; n,i:byte; f:text; function divizor(k:word):word; var d:word; begin d:=2; while(k mod d<>0) do d:=d+1; divizor:=d; end; begin assign(f,'date.in'); reset(f); readln(f,n); for i:=1 to n do begin read(f,k); s:=s+divizor(k); end; close(f); writeln(s); end.</pre>
<p>4. Pentru intensiv:</p> <p>Pentru fiecare element din vector din poziția i se verifică paritatea elementelor începând cu poziția i+1 și se afișează cele care au aceeași paritate cu a[i].</p> <p>Varianta C/C++</p> <p>Program intensiv</p> <pre>#include<fstream.h> unsigned int a[101],n,k; void main() { ifstream f("date.in"); f>>n; for(int i=1;i<=n;i++) f>>a[i]; f.close(); for(i=1;i<n;i++)</pre>	<p>Pentru neintensiv:</p> <p>Am folosit o funcție recursivă pentru aflarea valorii expresiei E(x).</p> <p>Varianta PASCAL</p> <p>Program intensiv</p> <pre>program bac; var n,i,j:byte; a:array [1..101] of word; f:text; t:boolean; begin assign(f,'date.in'); reset(f); readln(f,n); for i:=1 to n do read(f,a[i]);</pre>


```

for(int j=i+1;j<=n;j++)
    if((a[i]%2==a[j]%2)
        {cout<<a[i]<<" "<<a[j]<<endl;
          k=1;
        }
    if(!k)
        cout<<"0";
}

```

Program neintensiv

```

#include<iostream.h>
#include<math.h>

unsigned int n;
int a[10],x;

long suma(int k)
{
    if(k==n)
        return a[k];
    else
        return a[k]*pow(x,n-k)+suma(k+1);
}

void main()
{
    do{
        cout<<"n="; cin>>n;
    }while(n>9);
    for(int i=0;i<=n;i++)
    {
        cout<<"a["<<i<<"]=";
        cin>>a[i];
    }
    do{
        cout<<"x="; cin>>x;
    }while(x<-9||x>9);
    cout<<suma(0);
}

```

```

close(f);
for i:=1 to n-1 do
    for j:=i+1 to n do
        if (a[i] mod 2= a[j] mod 2) then
            begin
                writeln (a[i],' ', a[j]);
                t:=true;
            end;
        if t=false then
            writeln('0');
end.

```

Program neintensiv

```

Program bac;
var a:array[0..9] of integer;
    n,i:byte;
    x:integer;
function suma(k:byte):longint;
var p:longint;
begin
    if k=n then
        suma:=a[n];
    else
        begin
            p:=1;
            for i:=1 to n-k do
                p:=p*x;
            suma:=a[k]*p+suma(k+1);
        end;
end;
begin
    repeat
        write('n=');
        readln(n);
    until(n<=9);
    for i:=0 to n do
        begin
            write('a[' , i, ']=');
            readln(a[i]);
        end;
    write('x=');
    readln(x);
    writeln(suma(0));
end.

```

Varianta <52>

SUBIECTUL I

1. a

2. a) 2

b) 13 48 625 19 (un numar de forma $2 \cdot 4 + x$, un numar de forma $5 \cdot 4 + y$ si inca doua numere)

c) citeste n

d <- 0

c <- 0

i <- 1

executa

citeste x

```

cat timp x%2=0 executa
  x <- [x/2]; d <- d+1
cat timp x%5=0
  x <- [x/5]; c <- c+1
  i<-i+1
cat timp i<n
daca c<d atunci
  scrie c
altfel
  scrie d

```

```

d) #include<iostream.h>
int n,d,c,i,x;
void main()
{
  printf("%d n="); //cout<<"n=";
  scanf("%d",&n); //cin>>n;
d=0;
c=0;
for(i=1; i<=n; i++)
{
  printf("%d x="); //cout<<"x=";
  scanf("%d",&x); //cin>>x;
  while(x%2==0)
  {
    x=x/2;
    d++;
  }
  while(x%5==0)
  {
    x=x/5;
    c++;
  }
}
if(c<d)
  printf("%d",c); //cout<<c;
else
  printf("%d",d); //cout<<d;
}

```

```

Program bac;
var n,d,c,i,x: integer;
begin
  Write('n= '); read(n);
  d:=0; c:=0;
  for i:=1 to n do
  begin
    Write('x= '); read(x);

    While x mod 2 =0 do
    begin
      x:=x div 2;
      d:=d+1;
    end;
    While x mod 5=0 do
    begin
      x:=x div 5;
      c:=c+1;
    end;
  end;
  if c<d then
    Write(c)
  else
    Write(d);
end.

```

SUBIECTUL II

1. a)

3. b)

4. Pentru intensiv se șterge primul nod al listei

5. Pentru intensiv

Se citesc **n** cuvinte și pentru fiecare cuvânt citit se verifică dacă conține cel puțin o consoană, în caz contrar se numără cuvântul respectiv. S-a folosit un vector **b[]** care a fost inițializat cu vocale.

Pentru neintensiv 1 2 5

Pentru neintensiv

Se desparte șirul în cuvinte folosind separatorul spațiu. Se folosește un vector **b[]** în care este copiat primul caracter din fiecare cuvânt.

Varianta C/C++

```

Program intensiv
#include <iostream.h>
#include <string.h>

```

Varianta PASCAL

```

Program intensiv
program bac;
type litere=set of 'a'..'z';

```

```
void main ( )
{char s[101], b[]="aeiou";
 int n, i, nr=0;
 printf("%d n="); //cout<<"n=";
 scanf("%d",&n); //cin>>n;
 for (i=1; i<=n; i++)
 {printf("s= "); //cout<<"s=";
 scanf("%c",s); //cin>>s;
 int t=1;
 for(int j=0;j<strlen(s)&&t;j++)
 if(strchr(b,s[j])==0)
 t=0;
 if(t)
 nr++;
 }
 printf("%d",nr); //cout<<nr;
}
```

```
var voc:litere;
 s:string[20];
 i,n,j:byte;
 t:boolean;
 nr:byte;
begin
 voc=['a','e','i','o','u'];
 write('n=');readln(n);
 nr:=0;
 for j:=1 to n do
 begin
 write('s=');readln(s);
 t:=true;
 for i:=1 to length(s) do
 begin
 if not (s[i] in voc) then
 t:=false;
 end;
 if t=true, then
 nr:=nr+1;
 end;
 writeln(nr);
end.
```

Varianta C/C++

Program neintensiv

```
#include <iostream.h>
#include <string.h>
void main ( )
{char a[101], b[50], *p;
 printf("Dati sirul a \n ");
//cout<<"Dati sirul a"<<endl;
 gets(a); //cin.get(a,100);
 b[0]=0;
 p=strtok(a," ");
 strcat(b, p, 1);
 while (p)
 {p=strtok(NULL, " ");
 if(p)
 strcat(b,p,1);
 }
 puts(b); //cout<<b;
}
```

Varianta PASCAL

Program neintensiv

```
program bac;
 var a:string[100];
 b:string[50];
 i,k:byte;
begin
 write('Dati sirul a');
 readln(a);
 k:=0;
 i:=1;
 while (a[i]=' ') and (length(a)<>0)
 do
 delete(a,i,1);
 i:=length(a);
 while ( a[i]=' ') and
 (length(a)<>0) do
 begin
 delete(a,i,1);
 i:=i-1;
 end;
 b:=copy(a,1,1);
 for i:=2 to length(a) do
 if a[i]=' ' then
 b:=b+a[i+1];
 writeln(b);
end.
```

SUBIECTUL III

1. 120

2. $f(4)=10$ respectiv $f(100)=5050$

3. Pentru intensiv:

Se construiește vectorul caracteristic $v[]$ pentru a determina frecvența de apariție a unei cifre.

Pentru neintensiv:

Se construiește un vector caracteristic $v[]$ și se verifică dacă este format din

Se determină numărul maxim de cifre distincte pentru fiecare număr din fisier. Se afișează numărul corespunzător acestui maxim

componente distincte..

Varianta C/C++

a) Pentru intensiv

```
void dist(unsigned long a, int &b)
{
    int v[10];
    for (int i=0;i<=9;i++)
        v[i]=0;
    while (a!=0)
        {v[a%10]++; a=a/10;}
    for (i=0;i<=9;i++)
        if (v[i]!=0)
            b++;
}
```

b)

```
#include<iostream.h>
#include<fstream.h>
void dist(unsigned long a, int &b)
{int v[10];
for (int i=0;i<=9;i++)
    v[i]=0;
while(a!=0)
    {v[a%10]++;
    a=a/10;
}
for (i=0;i<=9;i++)
    if (v[i]!=0)
        b++;
}
void main ( )
{int n, i, max,b;
long int x[101];
ifstream f("date.in");
f>>n;
for (i=1; i<=n; i++)

    f>>x[i];
    max=1;
    for (i=1; i<=n; i++)
        {b=0;
        dist(x[i],b);
        if(b>max)
            max=b;
        }
    for (i=1; i<=n; i++)
        {b=0;
        dist(x[i],b);
        if(b==max)
            cout<<x[i]<<" ";
        }
}
```

Varianta PASCAL

Pentru intensiv

```
procedure dist(a:longint; var
b:byte);
var v:array[0..9] of byte;
    i:byte;
begin
    for i:=0 to 9 do
        v[i]:=0;
while (a<>0) do
begin
    v[a mod 10]:=v[a mod 10]+1;
    a:=a div 10;
end;

for i:=0 to 9 do
if (v[i]<>0) then
    b:=b+1;
end;

program bac2;
var n,i,max:byte;
    x:array[1..100] of longint;
    f:text;
    b:byte;
procedure dist(a:longint; var
b:byte);
var v:array[0..9] of byte;
    i:byte;
begin
    for i:=0 to 9 do
        v[i]:=0;
while (a<>0) do
begin
    v[a mod 10]:=v[a mod 10]+1;
    a:=a div 10;
end;
for i:=0 to 9 do
    if (v[i]<>0) then
        b:=b+1;
end;
begin
    assign(f,'date.in');
    reset(f);
    readln(f,n);
    for i:=1 to n do
        read(f,x[i]);
    max:=1;
    for i:=1 to n do
        begin
            b:=0;
            dist(x[i],b);
            if (b>max) then
                max:=b;
        end;
    for i:=1 to n do
```

Pentru neintensiv

a)

```
int dist (long int a)
{int w=0, i, v[9];
 while(a)
  {v[++w]=a%10;
   for (i=1; i<w; i++)
    if(v[w]==v[i])
     return 0;
   a/=10;
  }
return 1;
}
```

b)

```
#include<iostream.h>
int dist (long int a)
{
 int w=0,i,v[9];
 while(a)
 {
 v[++w]=a%10;
 for (i=1; i<w; i++)
  if (v[w]==v[i])
   return 0;
  a/=10;
 }
 return 1;
}

void main ( )
{int n, i, t=0;
 long int a[101];
 ifstream f ("date.in");
 f>>n;
 for (i=1; i<=n; i++)
 f>>a[i];
 for (i=1; i<=n; i++)
  if (dist(a[i]))

  {cout<<a[i]<<" ";
  }
 if (!t)
```

```
begin
 b:=0;
 dist(x[i],b);
 if (b=max) then
  write(x[i], ' ');
end;
```

end.

Pentru neintensiv

```
function dist(a:longint): boolean;
var v:array[1..10] of word;
 k,i:byte;
 t:boolean;
begin
 k:=0;t:=true;
 while(a<>0)do
 begin
 k:=k+1;

 v[k]:=a mod 10;
 for i:=1 to k-1 do
  if (v[k]=v[i]) then
   t:=false;
  a:=a div 10;
 end;
 if t=true then
  dist:=true
 else
  dist:=false;
 end;
```

```
program bac2;
var n,i,max:byte;
 x:array[1..100] of
 longint;
 a:array[1..101] of
 longint;
 f:text;
 t:boolean;
function dist(a:longint): boolean;
var v:array[1..10] of word;
 k,i:byte;
 t:boolean;
begin
 k:=0; t:=true;
 while(a<>0)do
 begin
 k:=k+1;
 v[k]:=a mod 10;
 for i:=1 to k-1 do
  if (v[k]=v[i]) then
   t:=false;
  a:=a div 10;
 end;
 if t=true then
  dist:=true
 else
  dist:=false;
 end;
begin
```

```
cout<<'0';
}
```

4. Pentru intensiv:

Folosim un *vector caracteristic*, care va reține frecvența de apariție a fiecărui număr. Dacă în vectorul caracteristic se găsesc numai elemente de 1 atunci elementele din tablou reprezintă o permutare.

Varianta C/C++

Program intensiv

```
#include<iostream.h>
#include<conio.h>
int a[100],v[100],n;
void main()
{do
{
printf("%d n="); //cout<<"n=";
scanf("%d",&n); //cin>>n;
}while(n<=0||n>100);
for(int i=1;i<=n;i++)
{ printf("a[%d]= ",i);
//cout<<"a["<<i<<"=";
scanf("%d",&n); //cin>>a[i];
v[a[i]]++;
}
int t=1;
for(i=1;i<=n&&t;i++)
if(v[i]!=1)

if(t) t=0;
printf("\n da"); //cout<<"da";
else
printf("\n nu"); //cout<<"nu";
getch();
}
```

Program neintensiv

```
#include<iostream.h>
#include<conio.h>
int n,a[100];
void main()
{ do
```

```
t:=false;
assign (f,'date.in');
reset(f);
readln(f,n);
for i:=1 to n do
read(f,a[i]);
for i:=1 to n do
if dist(a[i])=true then
begin
write(a[i], ' ');
t:=true;
end;
if t=false then
write('0');
end.
```

Pentru neintensiv:

Sortăm vectorul dat, calculăm rația progresiei aritmetice și verificăm dacă rația este constantă

Varianta PASCAL

Program intensiv

```
program p0006;
var a:array[1..101] of integer;
v:array[1..10000] of integer;
n,i:byte;
t:boolean;
begin
repeat
write('n=');
readln(n);
until (n<=100);
for i:=1 to n do
begin
write('a[' ,i, ']=');readln(a[i]);
v[a[i]]:=v[a[i]]+1;
end;
t:=true;
i:=1;
while (i<=n) and (t=true) do
begin
if(v[i]<>1) then
t:=false;
i:=i+1;
end;
if t=true then
write('da')
else
write('nu');
end.
```

Program neintensiv

```
program p0005;
var a: array[1..100] of integer;
n,i:byte;
t:boolean;
k, man:integer;
```

```

{printf("%d n="); //cout<<"n=";
 scanf("%d",&n); cin>>n;
}while(n<=0||n>100);
for(int i=1;i<=n;i++)
 { printf("a[%d]= ",i);
//cout<<"a["<<i<<"=";
scanf("%d",&a[i]); // cin>>a[i];
}
int t,man;
do
 {t=0;
  for(int i=1;i<n;i++)
  if(a[i]>a[i+1])
  {man=a[i];
   a[i]=a[i+1];
   a[i+1]=man;
   t=1;
  }

}while(t);
k=a[2]-a[1];
t=1;
for(i=2;i<n&&t;i++)
  if(a[i+1]-a[i]!=k)
  t=0;
if(t)
  printf("\n da"); //cout<<"da";
else
  printf("\n nu"); //cout<<"nu";
getch();
}

```

```

begin
repeat
write('n= ');
readln(n);
until(n<=100);
for i:=1 to n do
begin
write('a[' , i, ']=');
readln(a[i]);
end;
repeat
t:=false;
for i:=1 to n-1 do
if a[i]>a[i+1] then
begin
man:=a[i];
a[i]:=a[i+1];
a[i+1]:=man;

end;
until t=true;
k:=a[2]-a[1];
t:=true;
i:=2;
while (i<n) and (t=true) do
begin
if a[i+1]-a[i]<>k then
t:=false;
i:=i+1;
end;
if t=true then
write('da')
else
write('nu');
end.

```

Varianta <53>

SUBIECTUL I

1. c

2. a) 13

b) 8244

b)

```

citeste x
z←0
p←1
cat timp (x≠0) executa
  c←x%10
  daca (c%2≠0) atunci
    z←z+c*p
    p←p*10
  x←[x/10]
scrie z

```

d)) Varianta C/C++

```

#include<iostream.h>
void main()

```

Varianta PASCAL

```

program bac53;
var x,z,p,c:integer;

```

```

{
    int x,z=0,p=1,c;
    cout<<"dati x "; cin>>x;
    do
    {
        c=x%10;
        if(c%2!=0)
            {
                z=z+c*p;
                p=p*10;
            }
        x=x/10;
    }while(x!=0);
    cout<<z;
}

```

```

begin
    z:=0;
    p:=1;
    writeln('dat x ');
    read(x);
    repeat
        c:=x mod 10;
        if (c mod 2) <> 0 then
            begin
                z:=z+c*p;
                p:=p*10;
            end;
        x:=x div 10;
    until (x=0);
    writeln(z);
end.

```

SUBIECTUL II

1. c
2. a
3. 12

4. Program intensiv

returneaza pozitia primei aparitii a caracterului a in sir adica 2.

5. Program intensiv

Program neintensiv

vectorul TATA : 3 4 4 0 2 3 6 6

descendentii nodului 3 : 1 6 7 8

Program neintensiv

Varianta C/C++

Varianta PASCAL

Program intensiv

```
#include<iostream.h>
typedef struct nod
{
    unsigned int info;
    nod *urm;
};
nod * prim;
unsigned int n,a,r;
void creare(nod * &prim)
{
    nod *ultim, *nou;
    prim=new nod;
    prim->info=a;
    prim->urm=NULL;
    ultim=prim;
    for(int i=2;i<=n;i++)
    {
        nou=new nod;
        nou->info=a+(i-1)*r;
        nou->urm=NULL;
        ultimo->urm=nou;
        ultim=nou;
    }
}
void afisare(nod* p)
{nod *q;
q=p;
while (q)
{
    cout<<q->info<<" ";
    q=q->urm;
}
}
void main()
{
    do{
        cout<<"n=";cin>>n;
    }while(n>20);
    do{
        cout<<"a="; cin>>a;
    }while(a>10);

    do{cout<<"r="; cin>>r;
    }while(r>10);
    creare(prim);
    afisare(prim);
}
```

Varianta C/C++

Pentru neintensiv

```
#include<iostream.h>
void main()
{
    unsigned int m,n,x,y,i,j ;
```

Program intensiv

```
program bac53;
type adresa=^Nod;
    nod=record
        info:word;
        urm:adresa;
end;
var prim:adresa;
    n,a,r:word;
procedure creare(var prim:adresa);
    var nou,ultim:adresa;
        i:word;
    begin
        new(prim);
        prim^.info:=a;
        prim^.urm:=nil;
        ultim:=prim;
        for i:=2 to n do
            begin
                new(nou);
                nou^.info:=a+(i-1)*r;
                nou^.urm:=nil;
                ultim^.urm:=nou;
                ultim:=nou;
            end;
        end;
    procedure afisare(p:adresa);
        var q:adresa;
        begin
            q:=p;
            while(q<>nil) do
                begin
                    write(q^.info,' ');
                    q:=q^.urm;
                end;
            end;
        begin
            repeat
                write('n=');
                read(n);
            until (n<=20);
            repeat
                write('a=');
                read(a);
            until (a<=10);
            repeat
                write('r=');
                read(r);
            until (r<=10);
            creare(prim);
            afisare(prim);
        end.
```

Varianta PASCAL

Pentru neintensiv

```
program bac53;
type mat=array [1..10,1..20] of
integer;
var m,n,i,j,x,y :word ;
```

```

int a[10][20],aux;
cout<<"m="; cin>>m;
cout<<"n=";cin>>n;
for(i=1;i<=m;i++)
    for(j=1;j<=n;j++)
    {
        cout<<"a["<<i<<","<<j<<"]=";
        cin>>a[i][j];
    }
cout<<"x="; cin>>x;
cout<<"y="; cin>>y;
for(j=1;j<=n;j++)
{
    aux=a[x][j];
    a[x][j]=a[y][j];
    a[y][j]=aux;
}

for(i=1;i<=m;i++)

    for(j=1;j<=n;j++)
    cout<<a[i][j]<<"

";

    cout<<endl;
}
}

```

```

aux:integer;
mat a[10,20];
begin
    write('m='); read(m);
    write('n');read(n);
    for i:=1 to m do
        forj:=1 to n do
            begin
                write('a[' ,i ,','j ',']=');
                read(a[i,j]);
            end;
            write('dati linia x');
read(x);
            write('dati linia
y');read(y);
            for j:=1 to n do
                begin
                    aux:=a[x][j];
                    a[y,j]:=aux;
                end;
            end;
            for i:=1 to m do
                begin
                    for j:=1 to n do
                        write(a[i,j];
                        writeln;
                    end;
                end;
            end;
end;

```

SUBIECTUL III

1. a

2. Program intensiv

6 zerouri

3. a) unsigned int cmmdc(unsigned int a, unsigned int b)

Program neintensiv

f(2138)=10; f(513)=0. aduna cifrele pare

3. a) function cmmdc(x:word; y:word):word;

b) Varianta C/C++

Pentru intensiv

```

#include<fstream.h>
unsigned int cmmdc(unsigned int a,
unsigned int b)
{
    while(a!=b)
        if(a>b)
            a=a-b;
        else
            b=b-a;
        return a;
}
void main()
{
    unsigned int n,i,lmax=0,l;
    unsigned long a[100];
    ifstream f("date.in");

```

Varianta PASCAL

Pentru intensiv

```

program bac53;
var n,i,lmax,l:word;
    a:array[1..100] of word;
    f:text;
function cmmdc(x:word;
y:word):word;
begin
    while(x<>y) do

        if(x>y) then
            x:=x-y
        else
            y:=y-x;
    end;
end;

```

```

f>>n;
for(i=1;i<=n;i++)
f>>a[i];
f.close();
i=1;
while(i<n)
    {if(cmmdc(a[i],a[i+1])==1)
    {l=2;
    i++;

while(cmmdc(a[i],a[i+1])==1
&& i<n)
    {
    l++;
    i++;
    }
    if(l>lmax)
    lmax=l;

i++;
}
cout<<"lax="<<lmax<<endl;

}

```

Pentru neintensiv:

```

#include<fstream.h>
int cmmdc(int a, int b)
{
while(a!=b)
if(a>b)
a=a-b;
else
b=b-a;
return a;
}
void main()
{
int n,i, a[100];

ifstream f("date.in");
f>>n;
for(i=1;i<=n;i++)
f>>a[i];
f.close();
for(i=1;i<n;i++)
if(cmmdc(a[i],a[i+1])==1)
cout<<" "<<a[i]<<" "<<a[i+1]<<"
";
}

```

```

begin
assign(f,'date.in');
reset(f);
readln(f,n);
for i:=1 to n do
read(f,a[i]);
close(f);
lmax:=0;
i:=1;
while(i<n) do
begin
if(cmmdc(a[i],a[i+1])=1) then
begin
l:=2;
i:=i+1;
while((cmmdc(a[i],a[i+1])=1)
and(i<n)) do
begin
l:=l+1;
end;
if(l>lmax) then lmax:=l;
end;
i:=i+1;
end;
writeln('lmax=',lmax);
end.

```

Pentru neintensiv:

```

program bac53;
type vector=array[1..20] of
integer;
var n,i:integer;
f:text;
a:vector;
function cmmdc(a:integer;
b:integer):integer;
begin
while(a<>b) do
if(a>b) then
a:=a-b
else
b:=b-a;
cmmdc:=a;
end;
begin
assign(f,'date.in');
reset(f);
readln(f,n);
for i:=1 to n do
read(f,a[i]);
close(f);
for i:=1 to n-1 do
if(cmmdc(a[i],a[i+1])=1) then
write('(',a[i],',',a[i+1],')');
end.

```

4. Pentru intensiv:

Varianta C/C++

```
#include<iostream.h>
void main()
{
    unsigned int n, i, man,j,a[100];
    cout<<"n="; cin>>n;
    for(i=1;i<=n;i++)
    {
        cout<<"a["<<i<<"]=";
        cin>>a[i];
    }
    for(i=1;i<=n;i++)
    cout<<a[i]<<" ";
    cout<<endl;

    for(i=1;i<n;i++)
    {
        man=a[1];
        for(j=1;j<n;j++)
            a[j]=a[j+1];
            a[n]=man;

            for(j=1;j<=n;j++)
                cout<<a[j]<<" ";
            cout<<endl;

    }
}
```

Pentru intensiv :

Varianta PASCAL

```
program bac53;
type vector=array[1..100] of
word;
var i,n,j,man:word;
    a:vector;
begin
    write('n='); read(n);
    for i:=1 to n do
        begin
            write('a[' ,i, '=');
            read(a[i]);
        end;
    for i:=1 to n do
        write(a[i], ' ');
        writeln;
    for i:=1 to n-1 do
        begin
            man:=a[1];
            for j:=1 to n-1 do
                a[j]:=a[j+1];
                a[n]:=man;
                for j:=1 to n do
                    write(a[j], ' ');
                    writeln;
            end;
        end.
end.
```

Varianta <54>

SUBIECTUL I

1. d

2. a) 26

b) 1357, orice număr de 4 cifre format doar din cifre impare

c)

citeste n (număr natural nenul)

$s \leftarrow 0$

```
┌─cât timp n>0 execută
├─c←n%10
├─dacă%2=0 atunci
│   ┌─p←1
│   └─i←2
├─execută
│   ┌─p←p*i
│   └─cât timp i≤c
│       ┌─s←s+p
│       └─
└─
┌─ n←[n/10]
```

d) Varianta C/C++

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
void main()
{
unsigned int n,s,p,c,i;
do
{scanf("%u",&n); //cin>>n;
} while (n<=0) ;
s=0;
while(n>0)
{
c=n%10;
if (c%2==0)
{
p=1;
for(i=2;i<=c;i++)
p=p*i;
s=s+p;
}
n=n/10;
}
printf("%u",s); //cout<<s;
}
```

Varianta PASCAL

```
program bac;
var n,s,p,c,i: word;
begin
readln(n);
s:=0;
while n>0 do
begin
c:=n mod 10;
if c mod 2 = 0 then
begin
p:=1;
for i:=2 to c do
p:=p*i;
s:=s+p;
end;
n:=n div 10
end;
writeln(s)
end.
```

SUBIECTUL II

1. b

2. a

3. k=7, l=4
4. Pentru neintensiv

k=7

5.

Pentru intensiv

2 4

Varianta C/C++

Program intensiv

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
void main()
{ int a[10][10],n,i,j,x,c1,c2,c3;
do
{
scanf("%d",&n); //cin>>n;
}while(n<4 || n>10);
do
{
scanf("%d",&x); //cin>>x;
}while(x<100 || x>999);
c3=x%10;
x=x/10;
c2=x%10;
c1=x/10;
for(i=0;i<n;i++)
for(j=0;j<n;j++)
a[i][j]=c2;
for(i=0;i<n;i++)
a[i][i]=c1;
}
```

Varianta PASCAL

Program intensiv

```
type mat=array[1..10,1..10] of
integer;
var a:mat;
n,i,j,x,c1,c2,c3:integer;
begin
repeat
readln(n);
until((n>=4) and (n<=10));
repeat
readln(x);
until((x>=100) and (x<=999));
c3:=x mod 10;
x:=x div 10;
c2:=x mod 10;
c1:=x div 10;
for i:=1 to n do
for j:=1 to n do
if i=j then
a[i,j]:=c1
else j:=c1
if i+j= n+1 then
a[i,j]:=c3
end.
```

```

for(i=0;i<n;i++)
    a[i][n-i-1]=c3;
for(i=0;i<n;i++)
{
    for(j=0;j<n;j++)
        printf("%d",a[i][j]);
//cout<<a[i][j]<<" ";
    printf("\n"); //cout<<endl;
}
}

```

```

else
    a[i,j]:=c2;
for i:=1 to n do
begin
    for j:=1 to n do
        write(a[i,j], ' ');
    writeln()
end
end.

```

Varianta C/C++

Program neintensiv

```

#include<fstream.h>
#include<conio.h>
void main()
{ int n,x,a[10][10],pc,uc,i,j;
  do
  {
    scanf("%d",&n); //cin>>n;
  }while(n<3 || n>10);
  do
  {
    scanf("%d",&x); //cin>>x;
  }while(x<10 || x>99);
  uc=x%10;
  pc=x/10;
  for(i=1;i<=n;i++)
    for(j=1;j<=n;j++)
      if(i%2==0)
        a[i][j]=uc;
      else
        a[i][j]=pc;
  for(i=1;i<=n;i++)
  {
    for(j=1;j<=n;j++)
      printf("%d",a[i][j]);
    //cout<<a[i][j]<<" ";
    printf("\n");
    //cout<<endl;
  }
}

```

Varianta PASCAL

Program neintensiv

```

type mat=array[1..10,1..10] of
integer;
var a:mat;
n,x,pc,uc,i,j:integer;
begin
  repeat
    readln(n);
  until((n>=3) and (n<=10));
  repeat
    readln(x);
  until((x>=10) and (x<=99));
  uc:=x mod 10;
  pc:=x div 10;
  for i:=1 to n do
    for j:=1 to n do
      if i mod 2=0 then
        a[i,j]:=uc;
      else
        a[i,j]:=pc;
  for i:=1 to n do
  begin
    for j:=1 to n do
      write(a[i,j], ' ');
    writeln()
  end
end.

```

SUBIECTUL III

2. Pentru neintensiv

1, respectiv 0

3. a)

Pentru intensiv

1 2 3 4 5 2 2 1 1 0

Varianta C/C++

```

void pal(unsigned long a,unsigned
long &b)
{
  unsigned long k=0, x, ok=0, i=0,
j=0, aa=a,inv;
  while(!ok)
  {
    x=aa;

```

Varianta PASCAL

```

Procedure prim(a:longint, var
b:longint);
begin
  var k,x,ok,i,ju,aa,inv:int;
  k:=0;ok:=0;i:=0;j:=0;aa:=a;
  while(ok=0)
  begin
    x:=aa; inv:=0;
    while(aa<>0) do

```

```

    inv=0;
    while(aa!=0)
    {
        inv=inv*10+aa%10;
        aa=aa/10;

    }
    if(inv==x)
        ok=1;
    if(k%2==0)
    {
        i++;
        aa=a-i;
    }
    else
    {
        j++;
        aa=a+j;
    }
    k++;
}
b=inv;
}

```

```

b) #include<fstream.h>
#include<conio.h>
void main()
{
    clrscr();
    unsigned long x,y,n,i;
    ifstream f("date.in");
    f>>n;
    for(i=0;i<=n-1;i++)
    {
        f>>x;
        pal(x,y);
        printf("%d",y);
        //cout<<y<<" ";
    }
    getch();
}

```

4.
Varianta C/C++
Pentru intensiv

```

a) #include<iostream.h>
#include<conio.h>
void main()
{
    clrscr();
    int p=0,n,i,j,k,x,v[100];
    do
    {
        scanf("%d",&n); //cin>>n;
    }while(n>100);
    for(i=0;i<n;i++)

```

```

begin
    inv:=inv*10+aa mod 10;
    aa:=aa div 10
end;
if inv=x then
    ok:=1;
if (k mod 2=0)
begin
    i:=i+1; aa:=a-i
end
else
begin
    j:=j+1; aa:=a+j
end;
k:=k+1
end;
b:=inv;
end;

```

```

var x,y,n,i:integer;
f:text;
begin
    assign(f,'date.in');
    reset(f);
    read(f,n);
    for i:=1 to n do
    begin
        read(f,x);
        pal(x,y);
        write(y,' ');
    end;
    close(n);
end.

```

Varianta PASCAL
Pentru intensiv

```

program bac;
type vector=array[0..100] of integer;
var p,n,i,j,k,x:integer;
    v:vector;
begin
    p:=0;
    repeat
        readln(n);
    until n<=100;
    for i:=0 to n-1 do

```

```

do
{
scanf("%d",&v[i]);
//cin>>v[i];
}while(v[i]>9999);

for(i=0;i<n-1;i++)
for(j=i+1;j<n;j++)
if(v[i]==v[j])
{
v[j]=v[n-1];
n--;
}
for(k=0;k<n;k++)
for(i=0;i<n-1;i++)
for(j=i+1;j<n;j++)
if(k!=j&& k!=i
&&v[k]==v[i]+v[j])
p++;
printf("%d",p); //cout<<p;
getch();}

```

```

repeat
readln(v[i]);
until v[i]<10000;
for i:=0 to n-2 do
for j:=i+1 to n-1 do
if v[i]=v[j] then
begin
v[j]:=v[n-1];
n:=n-1
end;
for k:=0 to n-1 do
for i:=0 to n-2 do
for j:=i+1 to n-1 do
if (k<>j) and (k<>i) and
(v[k]=v[i]+v[j]) then
p:=p+1;
writeln(p);
end.

```

b) Se parcurge mai întâi vectorul și se elimină elementele care se repetă, astfel încât în vector să rămână numai elemente distincte. În continuare se caută acele elemente ale vectorului care pot fi scrise ca sumă de alte două elemente care apar în tabloului.

Varianta C/C++

Program neintensiv

```

#include<fstream.h>
#include<conio.h>

void main()
{
int n,i,v[100],k=0;
do
{
scanf("%d",&n); //cin>>n;
}while(n>100 || n==0);
for(i=1;i<=n;i++)
{
scanf("%d",&v[i]);
//cin>>v[i];
}

for(i=2;i<=v[i-1]+v[i+1])
k++;
printf("%d",n); //cout<<k;
}

```

Varianta PASCAL

Program neintensiv

```

type vector=array[1..100] of integer;
var n,i,k:integer;

v:vector;
begin
k:=0;
repeat
readln(n);
until n<=100;
for i:=1 to n do
readln(v[i]);
for i:=2 to n-1 do
if v[i]=v[i-1]+v[i+1] then
k:=k+1;
writeln(k);
end.

```

Varianta <55>

SUBIECTUL I

1. a

2. a) 3

b) Varianta C/C++

```

#include<iostream.h>
void main()
{

```

Varianta PASCAL

```

program I2d;
var n,a,b,k,i:integer;
begin

```

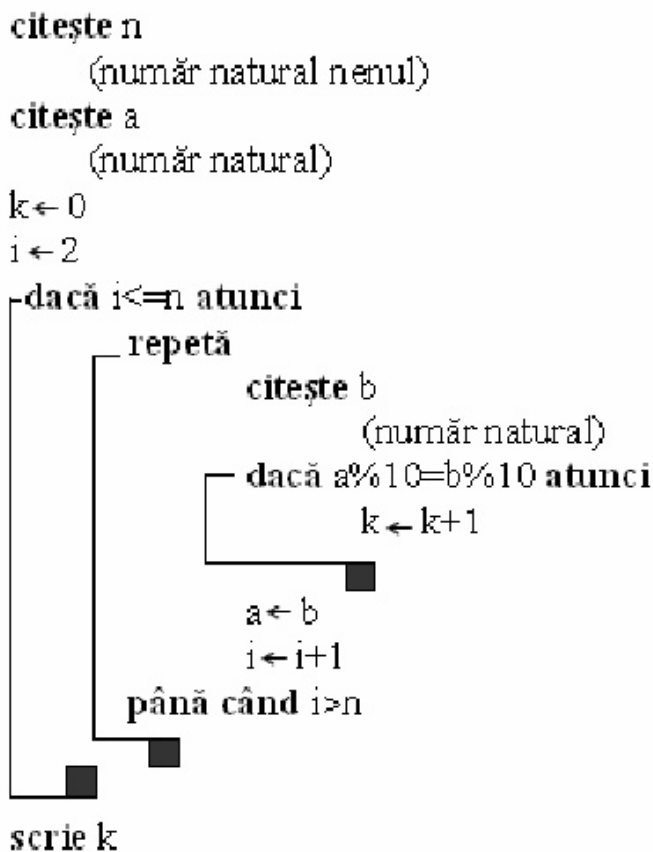


```
int n,a,b,k,i;
cin>>n>>a;
k=0;
for(i=2;i<=n;i++)
{
cin>>b;
if(a%10==b%10)
    k=k+1;
a=b;
}
cout<<"k=" <<k;
}
```

```
readln(n,a);
k:=0;
for i:=2 to n do
begin
    readln(b);
    if(a mod 10=b mod 10) then
        k:=k+1;
        a:=b;
end;
write(k);
end.
```

c) Se observă că programul afișează numărul de perechi de numere introduse pe poziții consecutive care au ultima cifră egală. Pentru a obține valoarea 4 putem introduce orice succesiune de numere naturale care au 4 perechi de numere introduse consecutiv cu ultima cifră egală. De exemplu:

d) 4 24 34 11 1 105 1905



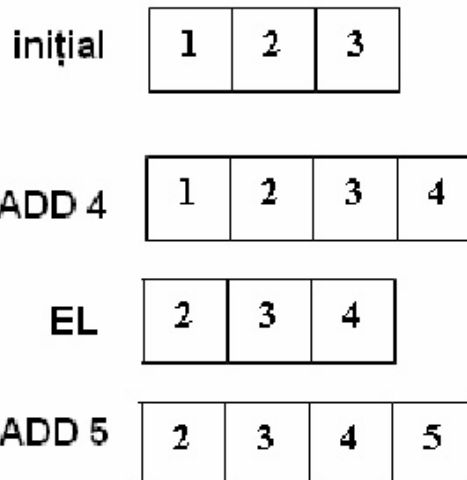
SUBIECTUL II

1. c pentru intensiv și c pentru neintensiv
2. Un graf orientat complet cu n noduri are n*(n-1) muchii. Numărul de submulțimi din mulțimea muchiilor este $2^{n*(n-1)}$.
Răspuns corect: a.

3.2 1

4. Pentru intensiv: Secvența șterge al doilea nod al unei liste liniare simplu înlănțuită.

Pentru neintensiv



5.

Varianta C/C++

Presupunând că, în mod obligatoriu șirul citit are și parte întreagă și parte zecimală, primul număr, a îl construim de la începutul șirului până

la caracterul . iar al doilea număr îl construim de la sfârșitul șirului spre punct, eliminând toate zerourile de la sfârșitul părții zecimale care, conform exemplului sunt ne semnificative.

Program intensiv

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
void main()
{
clrscr();
long a,b,i=0,p=1;
char s[10];
a=0;
b=0;
cout<<"s="; //printf("s=");
cin>>s; // scanf("%s",&s);
while(s[i]!='.' && s[i]!='\0')
{
a=a*10+(s[i]-'0');
i++;
}
i=strlen(s)-1;
while(s[i]=='0')
```

194

Varianta PASCAL

Program intensiv

```
program II5;
var a,b:longint;
s:string;
i,n,p:integer;
begin
R:=0;
b:=0;
write('s=');
readln(s);
i:=1;
n:=Length(s);
while(s[i]<>'.') do
begin
a:=a*10+(ord(s[i])-ord('0'));
i:=i+1;
end;
if(i<=n) then
i:=n;
while(s[i]='0') do
i:=i-1;
while(s[i]<>'.') do
```

```
i--;
while (s[i]!='.')
{
    b=b+(s[i]-'0')*p;
    p=p*10;

    i--;
}
cout<<a+b;
// printf("%d",a+b);
}
```

```
begin
b:=b+(ord(s[i])-ord('0'))*p;
p:=p*10;
i:=i-1;
end;
writeln(a+b);
end.
```

Varianta C/C++

Pentru neintensiv

```
#include<iostream.h>
#include<string.h>
void main()
{
char s[100];
int i=0;
cout<<"s="; //printf("s=");
cin.getline(s,100,'\n');
//gets(s);
while(s[i]!='\0')
{
    if(s[i]<'0' || s[i]>'9')

        strcpy(s+i,s+i+1);
    else
        i++;
}
cout<<s; // printf("%s",s);
}
```

Varianta PASCAL

Pentru neintensiv

```
program II5;
var s,s1,s2: string;
    n,i:integer;
begin
write('s=');
readln(s);
i:=1;
while(i<=Length(s)) do
begin
    if((s[i]<char('0')) or
(s[i]>char('9'))) then begin
        n:=Length(s);
        s:=copy(s,1,i-
1)+copy(s,i+1,n);
    end
    else
        i:=i+1;
end;
writeln(s);
end.
```

SUBIECTUL III

1. a

2. 20 pentru intensiv și 20 0 pentru neintensiv

3. a. pentru intensiv: Vom folosi un vector caracteristic c care va memora valoarea 0 dacă cifra i nu se găsește în numărul a și 1 dacă cifra i se găsește în numărul a. Parcurgând vectorul caracteristic vom de la 1 la 9 vom construi numărul b ce conține cifrele distincte ale numărului a. Dacăc[i]=1 atunci b=b*10+i.

pentru neintensiv: Vom sorta cifrele numărului a prin numărare. Elementul i al vectorului c memorează de câte apare

cifra i în numărul a. Construim b parcurgând vectorul c de la i=1 la 9 inserând în b cifra i de c[i] ori b=b*10+i.

Varianta C/C++

Pentru intensiv

```
void cifre(long a,long &b)
{
int c[10],i;

b=0;
for(i=1;i<=9;i++)
    c[i]=0;
while(a
```

Varianta PASCAL

Pentru intensiv

```
procedure cifre( a:longint; var
b:longint);
var c:array[1..10] of byte;

    i:integer;
begin
b:=0;
for i:=1 to 9 do
```

```

    {
    c[a%10]=1;
    a=a/10;
    }
for (i=1; i<=9; i++)
    if(c[i]==1)
        {
        b=b*10+i;
        }
}

```

Pentru neintensiv:

4. a) Pentru intensiv:

Am folosit subprogramul prim care verifică dacă valoarea transmisă în parametrul x reprezintă un număr prim. Vom citi numerele din fișier, rand pe rand în variabila x (citirea se finalizează în momentul în care s-a ajuns la sfârșitul fișierului). Variabilele a și b vor memora penultimul și respective ultimul număr prim citit din fișier, dacă acestea există. La citirea fiecărei valori se verifică dacă este număr prim și dacă noua valoare citită este diferită de ultima valoare memorată în

variabila a. În caz afirmativ variabila a primește preia noua valoare a numărului prim citit în x.

Varianta C/C++

Program intensiv

```

void cifre(long a, long &b)
{
int c[10], i;
b=0;
for(i=1; i<=9; i++)
    c[i]=0;
while(a)
    {
    c[a%10]=1;
    a=a/10;
    }
for(i=1; i<=9; i++)
    if(c[i]==1)
        {
        b=b*10+i;
        }
}

```

```

    c[i]:=0;
while (a<>0) do
begin
c[a mod 10]:=1;
a:=a div 10;
end;
for i:=1 to 9 do
    if(c[i]=1) then
        begin
        b:=b*10+i;
        end;
end;

```

Pentru neintensiv:

Pentru neintensiv:

Citirea valorilor s-a realizat testând finalul de fișier. Ultima valoare împărțită în variabila k. Pentru variabila b citim pentru a verifica dacă a existat sau nu măcar o valoare impară în fișier, iar la finalul programului afișăm rezultatul în funcție de valoarea acestei variabile.

Varianta PASCAL

Program intensiv

```

procedure cifre( a:longint; var
b:longint);
var c:array[1..10] of byte;
    i:integer;
begin
b:=0;
for i:=1 to 9 do
    c[i]:=0;
while (a<>0) do
begin
c[a mod 10]:=1;
a:=a div 10;
end;
for i:=1 to 9 do
    if(c[i]=1) then
        begin
        b:=b*10+i;
        end;
end;

```

Program neintensiv

```
void cifre(long a, long &b)
{
    int c[10], i, j;
    b=0;

    for(i=1; i<=9; i++)
    while(a)
        {
            c[a%10]++;
            a=a/10;
        }
    for(i=1; i<=9; i++)
        for(j=1; j<=c[i]; j++)
            {
                b=b*10+i;
            }
}
```

4. b. Citim numărul din fișerul **date.indac** în urma apelului funcției **cif(a,b)**, a=b atunci a este format numai din cifre distincte ordonate crescător.

Varianta C/C++

Program intensiv

```
#include<fstream.h>
void cifre(long a, long &b)
{
    int c[10], i;
    b=0;
    for(i=1; i<=9; i++)
        c[i]=0;
    while(a)
        {
            c[a%10]=1;
            a=a/10;
        }
    for(i=1; i<=9; i++)
        if(c[i]==1)
            {
                b=b*10+i;
            }
}

void main()
{
    long a, b;
    int n, i, gasit=0;
    ifstream f("date.in");
    f>>n;

    for(i=1; i<=n; i++)
        {
            f>>a;
            cifre(a, b);
            if(a==b)
                {
```

Program neintensiv

```
procedure cifre( a:longint; var
b:longint);
var c:array[1..10] of byte;
    i, j:integer;

begin
    for i:=1 to 9 do
        c[i]:=0;
    while (a<>0) do
        begin
            c[a mod 10]:=c[a mod 10]+1;
            a:=a div 10;
        end;
    for i:=1 to 9 do
        for j:=1 to c[i] do
            begin
                b:=b*10+i;
            end;
        end;
end;
```

4. b neintensiv. Citim numere din fișierul **date.in** Dac în urma apelul funcției **cifre(a,b)** a=b atunci numărul a are cifrele ordonate crescător și îl afișăm.

Varianta PASCAL

Program intensiv

```
procedure cifre( a:longint; var
b:longint);
var c:array[1..10] of byte;
    i:integer;

begin
    b:=0;
    for i:=1 to 9 do
        c[i]:=0;
    while (a<>0) do
        begin
            c[a mod 10]:=1;
            a:=a div 10;
        end;
    for i:=1 to 9 do
        if(c[i]=1) then
            begin
                b:=b*10+i;
            end;
        end;
end;

var a, b: longint;
    n, i, gasit: integer;
    f:text;

begin
    assign (f, 'date.in');
    reset(f);
    readln(f, n);

    for i:=1 to n do
        begin
            read(f, a);
            cifre(a, b);
```

```

        cout<<a<<" ";
        gasit=1;
    }

    }
    if(gasit==0)
        cout<<0;
}

```

4.b. Varianta C/C++

Program neintensiv

```

#include<fstream.h>
void cifre(long a,long &b)
{
    int c[10],i,j;
    b=0;
    for(i=1;i<=9;i++)
        c[i]=0;
    while(a)
    {
        c[a%10]++;
        a=a/10;
    }
    for(i=1;i<=9;i++)
        for(j=1;j<=c[i];j++)
            {
                b=b*10+i;
            }
}

void main()
{
    long a,b;
    int n,i,gasit=0;
    //FILE *f;
    //f=fopen("date.in","rt");
    ifstream f("date.in");
    f>>n;
    //fscanf(f,"%d",&n);
    for(i=1;i<=n;i++)
    {
        // fscanf(f,"%d",&a);
        f>>a;
        cifre(a,b);
        if(a==b)
            {
                //printf("%d ",a);
                cout<<a<<" ";
                gasit=1;
            }
    }
    if(gasit==0)
        cout<<0; // printf("0");
}

```

```

        if(a=b) then
            begin
                write(a,' ');
                gasit:=1;
            end;

        end;
    if(gasit=0) then
        write(0);

```

end.

4.b. Varianta Pascal

Program neintensiv

```

procedure cifre( a:longint; var
b:longint);
var c:array[1..10] of byte;
    i,j:integer;
begin
    b:=0;
    for i:=1 to 9 do
        c[i]:=0;
    while (a<>0) do
        begin
            c[a mod 10]:=c[a mod 10]+1;
            a:=a div 10;
        end;
    for i:=1 to 9 do
        for j:=1 to c[i] do
            begin
                b:=b*10+i;
            end;
        end;
end;

var a,b: longint;
    n,i,gasit: integer;
    f:text;
begin
    assign (f,'date.in');
    reset(f);
    readln(f,n);
    for i:=1 to n do
        begin
            read(f,a);
            cifre(a,b);
            if(a=b) then
                begin
                    write(a,' ');
                    gasit:=1;
                end;
        end;
    if(gasit=0) then
        write(0);
end.

```

end.

III.4

Deplasăm c =numărul cel mai mic pe care apoi îl deplasăm cu atâtea poziții câte cifre are numărul mai mare prin înmulțiri repetate cu 10 și adunăm la c numărul cel mai mare.

varianta C/C++

```
#include<iostream.h>
void main()
{
int a[101],b[101],n,i,copie;
long c[100];
cout<<"n="; //printf("n=");
cin>>n; //scanf("%d",&n);
for(i=1;i<=n;i++)
{
cout<<"a["<<i<<"]=";
cin>>a[i]; //scanf("%d",&a[i]);
}
for(i=1;i<=n;i++)

{cout<<"b["<<i<<"]=";
cin>>b[i]; //scanf("%d",&b[i]);
}
for(i=1;i<=n;i++)
{
if(a[i]<b[i])
{
c[i]=a[i];
copie=b[i];
while(copie)
{
c[i]=c[i]*10;
copie=copie/10;
}
c[i]=c[i]+b[i];
}
else
{
c[i]=b[i];
copie=a[i];
while(copie)
{
c[i]=c[i]*10;
copie=copie/10;
}
c[i]=c[i]+a[i];
}
}
for(i=1;i<=n;i++)
cout<<c[i]<<"
"; //printf("%d",c[i]);
}
```

Varianta Pascal

```
var a,b:array[1..100] of integer;
n,i,copie:integer;
c:array [1..100]of longint;
begin
write('n=');
readln(n);
for i:=1 to n do
begin
write('a[' ,i,']=');
readln(a[i]);
end;
for i:=1 to n do
begin
write('b[' ,i,']=');
readln(b[i]);
end;
for i:=1 to n do
begin
if(a[i]<b[i]) then
begin
c[i]:=a[i];
copie:=b[i];
while(copie<>0) do
begin
c[i]:=c[i]*10;
copie:=copie
div 10;
end;
c[i]:=c[i]+b[i];
end
else
begin
c[i]:=b[i];
copie:=a[i];
while(copie<>0) do
begin
c[i]:=c[i]*10;
copie:=copie
div 10;
end ;
c[i]:=c[i]+a[i];
end;
end;
for i:=1 to n do
write(c[i], ' ');
end.
```

SUBIECTUL I

1. b

2. a) 1020

b) 0897

c) Varianta C/C++

```
#include<iostream.h>
#include<stdio.h> //C
void main()
{
    unsigned int n,r;
    do {
        cin>>n;
        //scanf("%d",&n);
    } while(n<=0);
    r=0;
    do{
        r=(r*10+n%10)*10;
        n=n/100;
    } while(n>=10);
    cout<<r;
    //printf("%d",r);
}

```

d)

```
citeste n (numar natural)
r←0
cat timp n>=10
    r←(r*10+n%10)*10
    n←[n/100]
sfarsit cat timp

```

SUBIECTUL II

1. c

2. d

3. Frunzele arborelui sunt: 2 3 5 8

4. Varianta C/C++

(c>='a') && (c<='z')

5. Pentru intensiv, varianta C/C++

```
#include<iostream.h>
#include<stdio.h> //C
void main()
{
    int n,k,i,j,mat[23][23];
    do { cout<<"n=";
        cin>>n;
        //printf("n=");
        //scanf("%d",&n);
    } while((n<=2) || (n>=25));
    do { cout<<"k=";
        cin>>k;
        //printf("k=");
        //scanf("%d",&k);
    } while((k<=0) || (k>=n));
    for(i=0;i<n;i++)
        for(j=0;j<n;j++)
            if(i<k) if (j<k) mat[i][j]=1;
}

```

Varianta PASCAL

```
program unu;
var n,r:word;
begin
    repeat
        write('n= ');
        readln(n);
    until n>0;
    r:=0;
    repeat
        r:=(r*10+n mod 10)*10;
        n:=n div 100;
    until (n<10);
    write('r=',r);
end.

```

4. Varianta Pascal

(c>='a') AND (c<='z')

5. Pentru intensiv, varianta PASCAL

```
program unu;
type matrice=array[1..24,1..24] of integer;
var n,k,i,j:integer;
    mat:matrice;
begin
    repeat
        write('n= ');
        readln(n);
    until((n>2) and (n<25));
    repeat
        write('k= ');
        readln(k);
    until((k>0) and (k<n));
    for i:=1 to n do
        for j:=1 to n do
            if(i<=k) then if (j<=k)

```



```

        else mat[i][j]=2;
    else if(j<k) mat[i][j]=3;
        else mat[i][j]=4;
for(i=0;i<n;i++)
{
    for(j=0;j<n;j++)
        cout<<mat[i][j]<<" ";
    //printf("%d ",mat[i][j])
    cout<<endl;
    //printf('\n');
}
}

```

5. Pentru neintensiv, varianta C/C++

```

#include<iostream.h>
#include<stdio.h>//C
void main()
{
    int n,k,i,j,mat[23][23];
    do { cout<<"n=";
        cin>>n;

        //printf("%d\n",n);
        //scanf("%d",&n);
    } while((n<=2) || (n>=25));
    do { cout<<"k=";
        cin>>k;
        //printf("k=");
        //scanf("%d",&k);
    } while((k<=0) || (k>=n));
for(i=0;i<n;i++)
for(j=0;j<n;j++)
if(j<k) mat[i][j]=1;
else mat[i][j]=2;
for(i=0;i<n;i++)
{
for(j=0;j<n;j++)
cout<<mat[i][j]<<" ";
cout<<endl;
}
}
}

```

```

        then mat[i,j]:=1
        else mat[i,j]:=2
    else if(j<=k) then mat[i,j]:=3
        else mat[i,j]:=4;
for i:=1 to n do
begin
for j:=1 to n do
write(mat[i,j], ' ');
writeln;
end;
end.

```

5. Pentru neintensiv, varianta PASCAL

```

program unu;
type matrice=array[1..24,1..24] of
integer;
var n,k,i,j:integer;
mat:matrice;
begin
repeat write('n= ');
readln(n);
until((n>2) and (n<25));
repeat
write('k= ');
readln(k);
until((k>0) and (k<n));
for i:=1 to n do
for j:=1 to n do
if(j<=k) then mat[i,j]:=1
else mat[i,j]:=2;
for i:=1 to n do
begin
for j:=1 to n do
write(mat[i,j], ' ');
writeln;
end;
end.

```

SUBIECTUL III

1. b

2. 43 pentru intensiv și 5 pentru neintensiv

3. Pentru intensiv, varianta C++

```

#include<fstream.h>
int x[10000];
void main()
{
int i,j,nr;
ifstream in("numere.txt");
ofstream out("pare.txt");
while(in>>nr)
x[nr]++;
for(i=2;i<10000;i+=2)
if(x[i]!=0)

for(j=1;j<=x[i];j++)
out<<i<<" ";
in.close();
out.close();
}

```

3. Pentru neintensiv, varianta C++

```

#include<fstream.h>
void main()
{
int nr;
ifstream in("numere.txt");
ofstream out("pare.txt");
while(in>>nr)
if(nr%2==0) out<<nr<<" ";
in.close();
out.close();
}

```

3. Pentru intensiv, varianta C

```
#include<stdio.h>
int x[10000];
void main()
{
    int i,j,nr;
    FILE *in, *out;
    in=fopen("numere.txt","r");
    out=fopen("pare.txt","w");
    while(!feof(in))
    {
        fscanf(in,"%d",&nr);
        x[nr]++;
    }
    for(i=2;i<10000;i+=2)
        if(x[i])
            for(j=1;j<=x[i];j++)
                fprintf(out,"%d ",i);
    fclose(in);
    fclose(out);
}
```

3. Pentru intensiv, varianta Pascal

```
program unu;
type    tip_nou=1..9999;
vector=array[1..9999] of
tip_nou;
var     f,g:text;
        nr,i,j:integer;
        x:vector;
begin
    assign(f,'numere.txt');
    reset(f);
    while(not eof(f)) do
    begin
        read(f,nr);
        x[nr]:=x[nr]+1;
    end;
    close(f);
    assign(g,'pare.txt');
    rewrite(g);
    for i:=2 to 9999 do
    if((x[i]>0)and(i mod 2=0))
        then for j:=1 to x[i] do
            write(g,i,' ');
        close(g);
end.
```

4. a) Varianta C/C++

```
void numar(int x, int &nrp)
```

4. b)

```
void main()
{
    int a,b,nrpa=0,nrpb=0,aux;
    cout<<"a = ";
    cin>>a;
    cout<<"b = ";
    cin>>b;

    if(b<a){ aux=a;
            a=b;
            b=aux;
        }
}
```

3. Pentru neintensiv, varianta C

```
#include<stdio.h>
void main()
{
    int nr;
    FILE *in, *out;
    in=fopen("numere.txt","r");
    out=fopen("pare.txt","w");
    while(!feof(in))
    {
        fscanf(in,"%d",&nr);
        if(nr%2==0)
            fprintf(out,"%d ",nr);
    }
    fclose(in);
    fclose(out);
}
```

3. Pentru neintensiv, varianta Pascal

```
program unu;
var     f,g:text;
        nr:integer;
begin
    assign(f,'numere.txt');
    reset(f);
    assign(g,'pare.txt');
    rewrite(g);
    while(not eof(f)) do
    begin
        read(f,nr);
        if(nr mod 2=0)
            then write(g,nr,' ');
    end;
    close(f);
    close(g);
end.
```

4. a) Varianta PASCAL

```
Procedure numere(x:integer,Var
nrp:integer)
```

4. b)

```
var a,b,nrpa,nrpb,aux:integer;
begin
    nrpa:=0;
    nrpb:=0;
    write('a= ');
    readln(a);

    write('b= ');
    readln(b);
    if(b<a) then
    begin aux:=a;
```

```

numar (a, nrpa);
numar (b, nrpb);
if ((nrpb-nrpa)>0) cout<<"DA";
else cout<<"NU";
}

```

```

a:=b;
b:=aux;
end;
numar (a, nrpa);
numar (b, nrpb);
if ((nrpb-nrpa)>0) then
write('DA')
else write('NU');
end.

```

Varianta <57>

SUBIECTUL I

1. d

2.

a)6

b) Oricare dintre numerele naturale : 63, 70, 77, 91,...

d) citește n (număr natural)

```

q ← 1
i ← 1
- - dacă i < [n/i] atunci
  - - repetă
    - - dacă n%i=0 atunci
      q ← q+i
    - - i ← i+3
  - - până când i ≤ [n/i]
- - scrie q

```

c)

Varianta C/C++

```

#include<iostream.h>
void main()
{unsigned int n,q=1,i=1;
cin>>n;
while (i<n/i)
{ if(n%i==0) q+=i;
i+=3;
}
cout<<q;
}

```

Varianta PASCAL

```

Program Varianta57;
var n,q,i: integer;
begin
readln(n);
q:=1;i:=1;
while ( i<n div i ) do
begin
if (n mod i=0) then q:=q+i;
i:=i+3;
end;
writeln(q);
end.

```

SUBIECTUL II

1. c

2. a

3. Nodul 5

4.

Varianta C/C++

```

#include <iostream.h>
#include <string.h>

```

Varianta PASCAL

```

Program subiect2_4;
uses Strings;

```

```
void main()
{ char sir[6],c;
  strcpy(sir,"aeiou"); // Se
  inițializează variabila

sir cu valoarea „aeiou”
  cin>>c;
  if(strchr(sir,c) ||
strchr(strupr(sir),c))
/*expresia strchr(sir,c) ||
strchr(strupr(sir) întoarce valoarea
1 dacă și numai dacă variabila c
este o vocală cu majuscule sau cu
minusculă*/
cout<<"Da";else cout<<"Nu";
}
```

5.

Varianta C/C++ //Pentru intensiv

```
#include <iostream.h>
int A[25][25],a,n,b;
void main()
{int i,j;
cin>>n>>a>>b;
for(i=1;i<a;i++)
  {for(j=1;j<b;j++) A[i][j]=1;
  for(j=b+1;j<=n;j++) A[i][j]=2;
  }
for(i=a+1;i<=n;i++)
  {for(j=1;j<b;j++) A[i][j]=3;
  for(j=b+1;j<=n;j++) A[i][j]=4;
  }
for(i=1;i<=n;i++)
  { for(j=1;j<=n;j++)
    cout<<A[i][j]<<' ';
    cout<<endl;
  }
}
```

//Pentru neintensiv

```
#include <iostream.h>
int A[25][25],a,n;
/*Matricea A este variabilă globală,
prin urmare elementele sale sunt
inițializate cu 0. (Elementele
liniei a vor fi toate nule, așadar
nu mai este necesară atribuirea
A[a][j]=0, pentru j de la 1 la n).*/
void main()
{int i,j;
cin>>n>>a;
for(i=1;i<a;i++)
```

```
const sir='aeiouAEIOU';
{constanta sir reține vocalele
scrise cu majuscule și cu minusculă}
var c:char;
begin
readln(c);
{Funcția pos returnează poziția
primei apariții a caracterului c în
șirul sir sau valoarea 0 dacă c nu
se regăsește în sir}
if (pos(c,sir)<>0) then
  writeln('Da')
else
  writeln('Nu')
end.
```

Varianta PASCAL {Pentru intensiv}

```
Program Varianta_57_5;
var Mat:array[1..25,1..25] of
integer;
  a,n,i,j,b:integer;
begin
readln(n,a,b);
for i:=1 to a-1 do
  begin
  for j:=1 to b-1 do
Mat[i,j]:=1;
  for j:=b+1 to n do
Mat[i,j]:=2;
  end;
for i:=a+1 to n do
  begin
  for j:=1 to b-1 do Mat[i,j]:=3;
  for j:=b+1 to n do Mat[i,j]:=4;
  end;
for i:=1 to n do
  begin
  for j:=1 to n do
write(Mat[i][j],' ');
writeln;
  end;
end.
```

//Pentru neintensiv

```
Program Varianta57;
var Mat:array[1..25,1..25] of
integer;
  a,n,i,j:integer;
begin
readln(n,a);
for i:=1 to a-1 do
  for j:=1 to n do
Mat[i,j]:=1;
for j:=1 to n do
Mat[a,j]:=0;
for i:=a+1 to n do
```

```

for(j=1;j<=n;j++)
  A[i][j]=1;
for(i=a+1;i<=n;i++)
  for(j=1;j<=n;j++)
    A[i][j]=2;
for(i=1;i<=n;i++)
{ for(j=1;j<=n;j++)
  cout<<A[i][j]<<' ';
  cout<<endl;
} }

```

```

for j:=1 to n do
  Mat[i,j]:=2;
for i:=1 to n do
  begin
    for j:=1 to n do
      write(Mat[i][j], ' ');
    writeln;
  end;
end.

```

SUBIECTUL III

1. a

2. 3112 pentru **intensiv**, respectiv 2213 pentru **neintensiv**

3. Pentru intensiv:

Programul preia din fișierul de intrare cele n numere și le reține în vectorul v . Se formează vectorul **sume**, cu elemente distincte, ordonate strict crescător, fiecare element al acestui vector reprezentând suma a două elemente din v . Se calculează suma s dintre fiecare al i -lea element al vectorului v și toate elementele ce-l succed ca poziție în vector (elementele de rang $i+1, i+2, \dots, n$). Se caută în vectorul **sume** valoarea s , inserându-se în vector pe poziția corespunzătoare. (Se folosește metoda de sortare prin inserție).

Varianta C/C++

```

#include <fstream.h>
int
n,v[101],i,j,sume[5050],m,s,N,gasit,k;
void main()
{fstream f("intrare.txt",ios::in);

  fstream g("iesire.txt",ios::out);
  f>>n;
  for(i=1;i<=n;i++)
    f>>v[i];
  sume[1]=v[1]+v[2];
  for(i=1;i<n;i++)
    for(j=i+1;j<=n;j++)
      {s=v[i]+v[j];
        gasit=0; k=1;
        while(k<=N && gasit==0)
          {if(sume[k]==s) gasit=1;
            else
              if(sume[k-1]<s && s<sume[k])
                {gasit=1;
                  for(m=N+1;m>k;m--)
                    sume[m]=sume[m-1];
                  sume[k]=s; N++;
                }
              k++;
            }
          if(!gasit) {N++;sume[N]=s;}
        }
  for(i=1;i<=N;i++)
    g<<sume[i]<<' ';
  f.close(); g.close();
}

```

Varianta Pascal

```

Program varianta57;
var f,g:text;
    v,sume:array[0..100] of
integer;
    n,i,j,k,gasit,nn,m,s:integer;

begin
  assign (f,'intrare.txt');
  reset(f);
  assign (g,'iesire.txt');
  rewrite(g);
  readln(f,n);
  for i:=1 to n do
    read(f,v[i]);
  sume[1]:=v[1]+v[2];nn:=1;
  for i:=1 to n-1 do
    for j:=i+1 to n do
      begin
        k:=1;gasit:=0;
        while(gasit=0) and (k<=N) do
          begin
            if(s=sume[k]) then
              gasit:=1
            else
              if (sume[k-1]<s) and
(s<sume[k]) then
                begin
                  gasit:=1;
                  for m:=nn+1 downto k
                    sume[m]:=sume[m-
1];
                  sume[k]:=s;
                end;
              nn:=nn+1;
            end;
          end;
        end;
      end;
end;

```

```

        k:=k+1;
    end;
    if(gasit=0) then
    begin
        nn:=nn+1;  sume[nn]:=s;
    end;
    end;
    for i:=1 to nn do
        write(g,sume[i], ' ');
    close(f); close(g);
    end.

```

Pentru neintensiv :

Programul preia din fişierul de intrare cele **n** numere și le reține în vectorul **v**. Se parcurge secvențial vectorul **v** de la primul până la penultimul element al său. Se scrie în fişierul de ieşire suma dintre fiecare al **i**-lea element al vectorului și toate elementele ce-l succed ca poziție în vector (elementele de rang **i+1, i+2, ..., n**).

Varianta C/C++

```

#include <fstream.h>
int n,v[101],i,j;
void main()
{fstream f("intrare.txt",ios::in);
 fstream g("iesire.txt",ios::out);
 f>>n;
 for(i=1;i<=n;i++)
     f>>v[i];
 for(i=1;i<n;i++)
     for(j=i+1;j<=n;j++)
         g<<v[i]+v[j]<<' ';
 f.close();
 g.close();
}

```

Varianta PASCAL

```

Program varianta57;
var f,g:text;
    v:array[1..100] of integer;
    n,i,j:integer;
begin
    assign (f,'intrare.txt'); assign
    (g,'iesire.txt');
    reset(f); rewrite(g);
    readln(f,n);
    for i:=1 to n do read(f,v[i]);
    for i:=1 to n-1 do
        for j:=i+1 to n do
            write(g,v[i]+v[j], ' ');
    close(f); close(g);
end.

```

4. a) Cel mai mic multiplu al numărului natural nenul **k**, mai mare sau egal decât numărul **a** se poate calcula cu ajutorul formulei : $\left(\left[\frac{a-1}{k}\right]+1\right) \cdot k$, unde se notează cu **[x]** partea

întreagă a numărului **x**.

Varianta C/C++

```

int multiplu(int a,int k)
{return ((a-1)/k+1)*k;
}

```

Varianta PASCAL

```

function
multiplu(a,k:integer):integer;
begin
multiplu:= ((a-1) div k + 1) *
k;
end;

```

b) Un număr **nr** ∈ **[x ,y]** dacă $x \leq nr \leq y$. Programul determină **m**=cel mai mic multiplu al numărului **z**, mai mare sau egal decât **y**. Prin urmare, numărul **m-z** va fi cel mai mare multiplu al lui **z**, mai mic sau egal decât **y**. Dacă numărul **m-z** ∈ **[x ; y]** atunci cel puțin un multiplu al lui **z** se află în intervalul dat.

Varianta C/C++

```

#include <iostream.h>
int x,y,z,m;
int multiplu(int a,int k)
{return ((a-1)/k+1)*k;
}
void main()

```

Varianta PASCAL

```

Program varianta57;
var x,y,z,m:integer;
function
multiplu(a,k:integer):integer;
begin
multiplu:= ((a-1) div k+1)*k;

```

```
{cin>>x>>y>>z;
m=multiplu(y,z)-z;
if ( m<=y && m>=x ) cout<<"DA";
else cout<<"NU";
}
```

```
end;
begin
readln(x,y,z);
m:=multiplu(y,z)-z;
if ( m<=y ) and ( m>=x ) then
writeln('DA')
else writeln('NU');
end.
```

Varianta <58>

SUBIECTUL I

1. c

2.

a) 1101

b) Oricare dintre numerele 25, 50, 75, 100

d) citește n (număr natural)

```
q ← 1
-- dacă n>0 atunci
-- repetă
--   dacă n%5=0 atunci
--     q ← q * 10
--   altfel
--     q ← q * 10 + 1
--   ■
--   n ← [n/5]
--   ■ până când n=0
--   ■
scrie q
```

c) Varianta C/C++

```
#include <iostream.h>
void main()
{unsigned int n,q=1;

cin>>n;
while (n>0)
{ if(n%5==0)
q*=10;
else
q=q*10+1;
n/=5;
}
cout<<q;
}
```

Varianta PASCAL

```
program varianta58;
var n,q:integer;
begin
readln(n);
q:=1;
while (n>0) do
begin
if n mod 5=0 then
q:=q*10
else
q:=q*10+1;
n:=n div 5;
end;
writeln(q);
end.
```

SUBIECTUL II

1. b

2. b

3.2

4.

Varianta C/C++

i!=j

5. Pentru intensiv

Se parcurge secvențial șirul citit. Se formează un nou șir de caractere doar cu literele din șirul inițial.

```
#include <iostream.h>
#include <string.h>
void main()
{char sir[256],SIR[256];
int i,nrlit=0;
cin.get(sir,256);
for(i=0;i<strlen(sir);i++)
    if((sir[i]>='a' &&
sir[i]<='z')||(sir[i]>='A' &&
sir[i]<='Z'))
        {SIR[nrlit]=sir[i];nrlit++;}
cout<<SIR<<endl;
}
```

Pentru neintensiv

Programul compară succesiv toate elementele șirului de caractere **sir** cu literele mici ale alfabetului englez. Variabila întreagă **nrlit** reține numărul de litere mici ale alfabetului englez, conținute de șirul de caractere. Se parcurge șirul de la prima poziție (**sir[0]**) până la ultima poziție (**sir[strlen(sir)-1]**). Se compară fiecare element **i** al șirului de caractere cu literele mici ale alfabetului englez, incrementându-se, dacă este cazul, variabila **nrlit**.

Varianta C/C++

```
#include <iostream.h>
#include <string.h>
void main()
{char sir[256];
int i,nrlit=0;
cin.get(sir,256);
for(i=0;i<strlen(sir);i++)
    if(sir[i]>='a' && sir[i]<='z')
nrlit++;
cout<<nrlit<<endl;
}
```

Varianta PASCAL

i<>j

```
program varianta58;
var sir,sirnou:string;
    nrlit,i:integer;
begin
readln(sir);
sirnou:='';
for i:=1 to length(sir) do
    if ((sir[i]>='a') and
(sir[i]<='z')) or ((sir[i]>='A')
and (sir[i]<='Z'))
        then sirnou:=sirnou+sir[i];
writeln(sirnou);
end.
```

Programul compară succesiv toate elementele șirului de caractere **sir** cu literele mici ale alfabetului englez. Variabila întreagă **nrlit** reține numărul de litere mici ale alfabetului englez, conținute de șirul de caractere. Se parcurge șirul de la prima poziție (**sir[1]**) până la ultima poziție (**sir[length(sir)]**). Se compară fiecare element **i** al șirului de caractere cu literele mici ale alfabetului englez, incrementându-se, dacă este cazul, variabila **nrlit**.

Varianta PASCAL

```
program varianta58;
var sir:string;
    nrlit,i:integer;
begin
readln(sir);
for i:=1 to length(sir) do
    if (sir[i]>='a') and
(sir[i]<='z') then nrlit:=nrlit+1;
writeln(nrlit);
end.
```

SUBIECTUL III

1. c

2. Pentru intensiv 33, respectiv 1111 pentru neintensiv

3. Pentru intensiv : Notăm în ordine, cele mai mici trei numere citite din fișier : **min1**, **min2** și **min3**. Se citește din fișier primul număr **x** și se inițializează **min1**, **min2** respectiv **min3** cu această valoare citită. Se citesc apoi succesiv din fișier celelalte numere naturale (

pînă la ârșitul fișierului), comparîndu-se numărul citit cu cele trei valori **min1** , **min2**, respectiv **min2**. In cazul în care se găsește o valoare mai mică decît **min1**, se translează celelalte valori minime : **min2**<-**min1**, respectiv **min3**<-**min2**.

```
#include <fstream.h>
int n,i,min1,min2,min3,x;

void main()
{fstream f("bac.txt",ios::in);
f>>x;
min1=x;min2=x;min3=x;
while(f>>x)
if(x<min1)

{min3=min2;min2=min1;min1=x;}
else
if (x<min2)
{min3=min2;min2=x;}
else
if(x<min3) min3=x;
cout<<min3<<' '<<min2<<'
'<<min1<<endl;
f.close();
}
```

```
program varianta58;
var f:text;

begin
x,min1,min2,min3:integer;
assign(f,'bac.txt');
reset(f);read(f,x);
min1:=x;min2:=x;min3:=x;
while not eof(f) do
begin
read(f,x);
if x<min1 then
begin
min3:=min2;min2:=min1;min1:=x;end
else
if x<min2 then begin
min3:=min2;min2:=x;end;
end;
writeln(min3,' ',min2,' ',min1);
end.
```

Pentru neintensiv : Notăm în ordine, cele mai mici douănumere citite din fișier : **min1** și **min2**.

Se citește din fișier primul număr **x** și se inițiază **min1**, respectiv **min2** cu această valoare citită. Se citesc apoi succesiv din fișier celelalte numere naturale (pînă la ârșitul fișierului), comparîndu-se numărul citit cu cele douăvalori **min1** , respectiv **min2**.

Varianta C/C++

```
#include <fstream.h>
int n,i,min1,min2,x;
void main()
{fstream f("bac.txt",ios::in);
f>>x;
min1=x;min2=x;
while(f>>x)
if(x<min1) min1=x;
else
if (x<min2) min2=x;
cout<<min1<<' '<<min2;
f.close();
}
```

Varianta PASCAL

```
program varianta58;
var f:text;
x,min1,min2:integer;
begin
assign(f,'bac.txt');
reset(f);
readln(f,x);
min1:=x;min2:=x;
while not eof(f) do
begin
read(f,x);
if x<min1 then min1:=x
else
end;if x<min2 then min2:=x;
writeln(min1,' ',min2);
end.
```

4.

a) Subprogramul **divizor** primește ca parametru de intrare variabila întregă **a**. Parametru de ieșire este variabila întregă **d**. Cel mai mare divizor propriu posibil al numărului **a** este $[a/2]$, valoare cu care este inițializată variabila **d**. Cu o structură **while** se parcurg toate numerele naturale mai mici sau egale cu $[a/2]$, pînă se întâlnește primul divizor al numărului **a**, mai mic sau egal cu $a/2$. Algoritmul este finit, deoarece chiar dacă numărul **a** este prim, deci nu are divizori proprii, cel puțin numul **1** va divide pe **a**($d=1$)

Varianta C/C++

```
void divizor(int a,int &d)
{d=a/2;
while (a%d!=0) d--;
```

Varianta PASCAL

```
procédure divizor(a:integer;var
d:integer);
begin
```

}

```
d:=a div 2;
while(a mod d <>0) do d:=d-1;
end;
```

b) In programul principal este apelat subprogramul **divizor**, care va întoarce prin intermediul parametrului de ieşire **d**, valoarea celui mai mare divizor al lui **x**, mai mic decât

x. Dacă acesta este diferit de **1**, înseamnă că numărul **x** are cel puțin un divizor propriu, prin urmare nu este prim.

Varianta C/C++

```
#include <iostream.h>
int x, d;
void divizor(int a, int &d)
{d=a/2;
 while (a%d!=0)
  d--;
}
void main()
{cin>>x;

 divizor(x, d);
 if (d==1) cout<<"DA";
 else     cout<<"NU";
}
```

Varianta PASCAL

```
program varianta58;
var x, d: integer;
procedure divizor(a: integer; var
d: integer);
begin
d:=a div 2;
while(a mod d <>0) do d:=d-1;
end;
begin
 readln(x);
 divizor(x, d);
 if d=1 then writeln('DA')
 else writeln('NU')
end.
```

Varianta <59>

SUBIECTUL I

1. b

2.

a) 2

b) Oricare dintre numerele 31, 32, 33, 320, ...

d) citește **n** (număr natural)

```
-- cât timp  $n \geq 10$  execută
  n ← [(n%100)/10] + [n/10]
-- ■
serie n
```

c)

Varianta C/C++

```
#include <iostream.h>
void main()
{unsigned long int n;
 cin>>n;
 do
  { n=(n%100)/10 + n/10;
  } while ( n>=10);
 cout<<n;
}
```

Varianta PASCAL

```
program varianta59;
var n: longint;
begin
 readln(n);
 repeat
  n:=(n mod 100) div 10 + n
div 10;
 until (n<10);
 writeln(n);
end.
```

SUBIECTUL II

1. a
 2. c
 3. 1; 2; 6; 7; 8
 4. Pentru intensiv:
Varianta C/C++
 i+j==11
 Pentru neintensiv:
Varianta C/C++
 i==9

Varianta PASCAL
 i+j=11

Varianta PASCAL
 i=9

5. In vectorul **nrap[10]** se va reține numărul de apariții a fiecărei cifre din text(**nrap[i]** reține numărul de apariții a cifrei **i**, $i=0,..,9$). Se parcurge șirul **sir**, secvențial. La fiecare apariție a cifrei **i**, se incrementează cu o unitate elementul **nrap[i]**. Variabila **max**, inițializată cu **0**, reține elementul maxim din tabloul **nrap** iar variabila **imax** reține poziția din vector a elementului maxim. Dacă nu sunt cifre în text, variabila **max** va rămâne inițializată cu valoarea **0**, iar la sfârșitul algoritmului se va afișa un mesaj corespunzător.

Varianta C/C++

```
#include <iostream.h>
#include <string.h>
int nrap[10],max,imax,i;
char sir[256],cifre[10];
void main()
{
    strcpy(cifre,"0123456789");
    cin.get(sir,256);
    for(i=0;i<strlen(sir);i++)
        if(strchr(cifre,sir[i]))
            nrap[(int)sir[i]-(int)'0']++;
    for(i=0;i<=9;i++)
        if(nrap[i]>max)
            {max=nrap[i];imax=i;}
    if(max==0) cout<<"NU"<<endl;
    else cout<<imax<<endl;
}
```

Varianta PASCAL

```
program varianta59;
var sir:string;
    nrap:array[0..9] of integer;
    i,imax,max,nr,err:integer;
begin
    readln(sir);
    imax:=0;max:=0;
    for i:=1 to length(sir) do
        begin
            val(sir[i],nr,err);
            if err=0 then
                nrap[nr]:=nrap[nr]+1;
        end;
    for i:=0 to 9 do
        if(max<nrap[i]) then
            begin
                max:=nrap[i];
                imax:=i;
            end;
    if(max=0) then writeln('NU')
    else writeln(imax);
end.
```

SUBIECTUL III

1. d
 2. Pentru intensiv 3452 , respectiv 828 pentru neintensiv
 3. Pentru intensiv : Notăm în ordine, cele mai mari trei numere citite din fișier : **max1**, **max2** și **max3**. Se citește din fișier primul număr **x** și se inițializează **max1**, **max2**, respectiv **max3** cu această valoare citită. Se citesc apoi succesiv din fișier celelalte numere naturale (pînă la sfârșitul fișierului), comparîndu-se numărul citit cu cele trei valori **max1**, **max2** respectiv **max3**.

Varianta C/C++

```
#include <fstream.h>
int n,i,max1,max2,x,max3;
```

Varianta PASCAL

```
Program varianta59;
var n,i,max1,max2,x,max3:integer;
```

```
void main()
{fstream f("bac.txt",ios::in);
f>>x;max1=x;max2=x;max3=x;
while(f>>x)
    if(x>max1)
{max3=max2;max2=max1;max1=x;}
    else
    if (x>max2) {max3=max2;max2=x;}
    else
    if (x>max3) max3=x;
cout<<(max1%10)*(max2%10)*(max3%10)%10;
f.close();
}
```

```
f:text;
begin
assign(f,'bac.txt');
reset(f);
read(f,x);
max1:=x;max2:=x;max3:=x;
while(not eof(f)) do
begin
if(x>max1) then
begin
max3:=max2;max2:=max1;max1:=x;
end
else
if (x>max2) then
begin max3:=max2;max2:=x; end
else
if (x>max3) then max3:=x;
end
read(f,x);
writeln((max1 mod 10)*(max2 mod
10)*(max3 mod 10) mod 10);
close(f);
end.
```

Pentru neintensiv : Notăm în ordine, cele mai mari două numere citite din fișier : **max1** și **max2**.

Se citește din fișier primul număr **x** și se inițializează **max1**, respectiv **max2** cu această valoare citită. Se citesc apoi succesiv din fișier celelalte numere naturale (pînă la sfârșitul fișierului), comparîndu-se numărul citit cu cele două valori **max1**, respectiv **max2**.

Varianta C/C++

```
#include <fstream.h>
int n,i,max1,max2,x;
void main()
{fstream f("bac.txt",ios::in);
max1=10001;max2=10000;
while(f>>x)
    if(x>max1) max1=x;
    else
    if (x>max2) max2=x;
cout<<(max1%10)*(max2%10)%10;
f.close();
}
```

Varianta PASCAL

```
program varianta59;
var f:text;
x,max1,max2:integer;
begin
assign(f,'bac.txt');
reset(f);read(f,x);
max1:=x;max2:=x;
while not eof(f) do
begin
read(f,x);
if x>max1 then max1:=x;
else
if x>max2 then max2:=x;
end;
writeln( ((max1 mod 10)*(max2 mod
10) )mod 10);
end.
```

4. Explicații

a) Subprogramul **divizor** primește ca parametru de intrare variabila întregă **a**. Parametru de ieșire este variabila întregă **d**. Cel mai mic divizor propriu posibil al numărului **a** este 2. Dacă numărul **a** este par, variabila **d** (**divizorul**) este inițializat cu 2. Dacă **a** este număr impar, atunci **a** nu poate avea divizori pari, prin urmare vom căuta cel mai mic divizor printre numerele impare iar **d** va fi inițializat cu 3. La fiecare pas, divizorul va fi incrementat cu 2. Algoritmul se oprește la găsirea celui mai mic divizor al lui **d**, mai mare decât 1. Dacă numărul **a** este prim, atunci **d** va fi egal cu **a**.

Varianta C/C++

```
void divizor(int a,int &d)
{if(a%2==0) d=2;
 else
  {d=3;
   while (a%d!=0)
    d+=2;
  }
}
```

Varianta PASCAL

```
procedure divizor(a:integer;var
d:integer);
begin
if a mod 2=0 then d:=2
else
begin
d:=3;
while(a mod d <>0) do d:=d+2;
end;
end
```

b) In programul principal este apelat subprogramul **divizor** , care va întoarce prin intermediul parametrului de ieşire **d**, valoarea celui mai mic divizor propriu al lui **a** (dacă **a** este număr compus) sau valoarea **a** .

Varianta C/C++

```
#include <iostream.h>
int x,d;

void divizor(int a,int &d)
{if(a%2==0) d=2;
 else
  {d=3;
   while (a%d!=0)
    d+=2;
  }
}

void main()
{cin>>x;
 divizor(x,d);
 if(d==x) cout<<"DA";
 else cout<<"NU";
}
```

Varianta PASCAL

```
program varianta59;
var x,d:integer;

procedure divizor(a:integer;var
d:integer);
begin
if a mod 2=0 then d:=2
else
begin
d:=3;
while(a mod d <>0) do d:=d+2;
end;
end;
begin
readln(x);
divizor(x,d);
if d=x then writeln('DA')
else writeln('NU')
end.
```

Varianta <60>

SUBIECTUL I

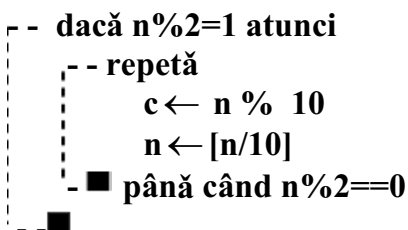
1. a

2.

a) 7

b) Oricare dintre numerele 32551 , 1234531, 257, 8851, 125....
d) citeşte n (număr natural)

c ← 10



serie c

c)

Varianta C/C++

```
#include <iostream.h>
void main()
{unsigned long int n;

unsigned int c;
cin>>n;
c=10;
while ( n%2==1)
{ c=n%10;
n=n/10;
} ;
cout<<c;
}
```

Varianta PASCAL

```
program varianta60;
var n:longint;
c:byte;

begin
readln(n);
c:=10;
while(n mod 2=1) do
begin
c:=n mod 10;
n:=n div 10;
end;
writeln(c);
end.
```

SUBIECTUL II

2. c

3. 1

4. 65 pentru **intensiv**, respectiv 75 pentru **neintensiv****5. Explicații**

În vectorul **nrap[10]** se va reține numărul de apariții a fiecărei litere mici din text (**nrap[i]** reține numărul de apariții a literei cu codul ASCII **i**, $i=97, \dots, 122$). Se parcurge șirul **sir**, secvențial. La fiecare apariție a unei litere cu codul ASCII **i**, se incrementează cu o unitate elementul **nrap[i]**. Variabila **max**, inițializată cu **0**, reține elementul maxim din tabloul **nrap** iar variabila **imax** reține poziția din vector a elementului maxim. Dacă nu sunt litere mici în text, variabila **max** va rămâne inițializată cu valoarea **0**, iar la sfârșitul algoritmului se va afișa un mesaj corespunzător.

Variabila **litere** de tip mulțime conține toate literele mici ale alfabetului englez. Funcția **ORD()** întoarce codul ASCII al caracterului primit ca parametru iar funcția **CHR()** întoarce caracterul al cărui cod ASCII este primit ca parametru.

Varianta PASCAL

```
program varianta60;
var sir:string;
nrap:array[97..122] of integer;
i,imax,max:integer;
litere:set of char;

begin
readln(sir);
imax:=0;max:=0;
litere:=[]; {Mulțimea litere se
inițializează cu mulțimea vidă}
for i:=ord('a') to ord('z') do
litere:=litere+[char(i)];
for i:=1 to length(sir) do
if sir[i] in litere then
nrap[ord(sir[i])]:=nrap[ord(sir[i])]+1;
for i:= ord('a') to ord('z') do
if(max<nrap[i]) then
```

Varianta C/C++

```
#include <iostream.h>
#include <string.h>
int nrap[123],măx,imax,i;
char sir[50];
void main()
{cin.get(sir,256);
for(i=0;i<strlen(sir);i++)
if(sir[i]<=(int)'z' &&
sir[i]>=(int)'a')
nrap[(int)sir[i]]++;
for(i=(int)'a';i<=(int)'z';i++)
if(nrap[i]>max)
{max=nrap[i];imax=i;}
if(max==0) cout<<"NU"<<endl;
else cout<<(char)imax<<endl;
}
```

```

begin
    max:=nrap[i];
    imax:=i;
end;
if(max=0) then writeln('NU')
else writeln(char(imax));
end.

```

SUBIECTUL III

1. d pentru **intensiv**, respectiv a pentru **neintensiv**
2. 1604 pentru **intensiv**, respectiv 3 pentru **neintensiv**
- 3.

O metodă pentru căutarea unui element într-un șir de elemente este căutarea secvențială. : căutarea pornește cu primul element și continuă până când elementul vizat este găsit sau șirul dat se termină. Complexitatea unui asemenea tip de algoritm este $O(N)$.

Atunci când se cunoaște găsetul de elemente este sortat, se poate crește viteza operației de căutare. Căutarea binară selectează elementul median din șir și compară valoarea sa cu cea de căutat. Dacă șirul este ordonat crescător iar valoarea elementului de căutat este mai mică decât valoarea mediană, atunci căutarea se află între marginea inferioară și valoarea mediană (în prima jumătate a șirului). În caz contrar, el se află în jumătatea superioară. În acest fel, cu o singură comparație, am renunțat la jumătate din spațiul de căutare. Algoritmul va continua apoi cu găsirea valorii mediane în noul spațiu de căutare (înjumătățit) și tot așa, până când se va putea da verdictul asupra existenței/non-existenței valorii căutate în șir. Complexitatea algoritmului de căutare binară este $O(\log_2 N)$.

Varianta C/C++

```

#include <fstream.h>
int n, i, x, nr, v[100], stg, dr, m;
void main()
{ fstream f("bac.txt", ios::in);
  cin >> x;
  while (f >> nr)
    { n++; v[n] = nr; }
  stg = 1; dr = n; m = (stg + dr) / 2;
  while (stg <= dr && v[m] != x)
    { if (x < v[m]) dr = m - 1;
      else stg = m + 1;
      m = (stg + dr) / 2;
    }
  if (x == v[m]) cout << "DA";
  else cout << "NU";
  f.close();
}

```

Varianta PASCAL

```

program varianta60;
var f: text;
    x, nr, m, stg, dr, n: integer;
    v: array[1..100] of integer;
begin
  assign(f, 'bac.txt'); reset(f);
  readln(x); nr := 0;
  while not eof(f) do
  begin
    n := n + 1; read(f, nr); v[n] := nr;
  end;
  stg := 1; dr := n; m := (stg + dr) div 2;
  while (stg <= dr) and (v[m] <> x) do
  begin
    if (x < v[m]) then dr := m - 1
    else stg := m + 1;
    m := (stg + dr) div 2;
  end;
  if (v[m] = x) then writeln('DA')
  else writeln('NU');
end.

```

4. Explicații

Subprogramul **radical** are doi parametri întregi: parametrul de intrare **a** și parametrul de ieșire **x**

Varianta C/C++

```
void radical(int a, int &x)
```

Varianta PASCAL

```
procedure radical(a: integer; var x: integer);
```

b) Prin intermediul parametrului de ieșire **x**, subprogramul întoarce partea întreagă a lui

\sqrt{a} , unde a este parametru de intrare. Dacă partea întreagă a lui \sqrt{n} (număr obținut prin apelul subprogramului **radical**, cu parametrii efectivi n și x) ridicată la pătrat este egală cu numărul n , atunci n este pătrat perfect.

Varianta C/C++

```
#include <math.h>
int n,x;
void radical(int a, int &x)
{ x=(int)sqrt(a);
}
void main()
{cin>>n;
 radical(n,x);
 if(n==x*x) cout<<"DA";
 else cout<<"NU";
}
```

Varianta PASCAL

```
Program varianta60;
Var n,x:integer;
procedure radical(a:integer; var
x:integer);
begin
x:=round(sqrt(a));
end;
begin
readln(n);
radical(n,x);
if (n=x*x) then
writeln('DA')
else writeln('NU');
end.
```

Varianta <61>

SUBIECTUL I

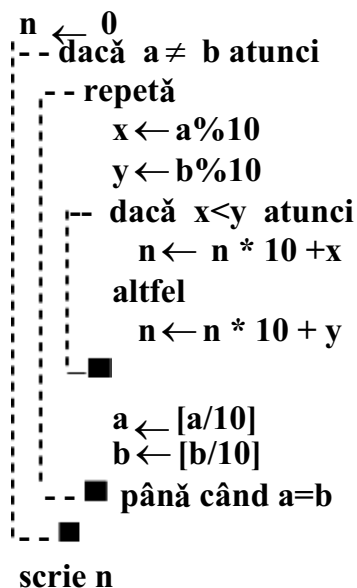
1. d

2.

a) 1303

b) 36 perechi

d) citește a,b (numere naturale care au același număr de cifre)



c)

Varianta C/C++

```
#include <iostream.h>
void main()
{unsigned long int a,b,n;
 unsigned int x,y;
 cin>>a>>b;
 n=0;
```

Varianta PASCAL

```
program varianta61;
var a,b,n:longint;
    x,y:byte;
begin
    readln(a,b);
    n:=0;
```



```
while (a!=b)
{
  x=a%10;
  y=b%10;
  if (x<y)
    n=n*10+x;
  else
    n=n*10+y;
  a/=10;
  b/=10;
}
cout<<n;
}
```

```
while (a<>b) do
begin
  x:= a mod 10;
  y:=b mod 10;
  if (x<y) then
    n:=n*10+x;
  else
    n:=n*10+y;
  a:=a div 10;
  b:=b div 10;
end;
writeln(n);
end.
```

SUBIECTUL II

La subiectul II, problema 5, apare în textul problemei condiția $2 < n \leq 100$. Matricea

$n \times n$ are elemente întregi. Elementul $a[1][1] = 3$. Pentru $n=100$,

Propun înlocuirea condiției cu $2 < n \leq 17$

1. d
2. c
3. 15 pentru intensiv, respectiv 99 pentru neintensiv
4. 4 pentru intensiv, respectiv f pentru neintensiv
5. Explicații

Varianta 1 de rezolvare

Se inițializează ultima linie a matricei cu elementele 1,2,...,n
 Se generează succesiv elementele de pe liniile n-1,n-2,...,1, după formula :
 $a[i][j]=a[i+1][j-1]+a[i+1][j]+a[i+1][j+1];$
 Elementele coloanei 0 și elementele coloanei n+1 sunt egale cu 0.

Varianta 2 de rezolvare

Se demonstrează prin inducție după n (ordinul matricei pătrate) că $a[i][j]=3^{n-i} * j$

Varianta C/C++

```
// Varianta 1 de rezolvare
#include <iostream.h>
int n,a[100][100],i,j;
void main()
{cin>>n;
for(j=1;j<=n;j++)
  a[n][j]=j;
for(i=n-1;i>=1;i--)
  for(j=1;j<=i;j++)
    a[i][j]=a[i+1][j-1]+a[i+1][j]+a[i+1][j+1];
for (i=1;i<=n;i++)
  { for (j=1;j<=n;j++)
    cout<<a[i][j]<<' ';
    cout<<endl;
  }
}

//Varianta 2 de rezolvare
#include <iostream.h>
#include <math.h>
```

Varianta PASCAL

```
{Varianta 1 de rezolvare}
program varianta61;
var n,i,j:integer;
    a:array[0..101,0..100] of
integer;
begin
  readln(n);
  {se bordeaza coloana n+1 si coloana
  0 cu 0}
  for i:=1 to n+1 do
    begin
      a[i,n+1]:=0; a[i,0]:=0;
    end;
  for j:= 0 to n+1 do a[n+1,j]:=0;
  for j:=1 to n do a[n,j]:=j;
  for i:=n-1 downto 1 do
    for j:=1 to i do
      a[i,j]:=a[i+1,j-1]+a[i+1,j]+a[i+1,j+1];
  for i:=1 to n do
```

```

long int n,a[100][100],i,j;
void main()
{cin>>n;
for (i=1;i<=n;i++)
  for (j=1;j<=i;j++)
    a[i][j]=(long int) pow(3,n-i)*j;
for (i=1;i<=n;i++)
  { for (j=1;j<=n;j++)
    cout<<a[i][j]<<' ';
    cout<<endl;
  }
cout<<(long int) pow(3,n-1);
}

```

```

for j:=1 to n do write (a[i,j], '
');
writeln;
end; end.
{Varianta 2 de rezolvare }
program varianta61;
var n,i,j:integer;
    a:array[0..101,0..100] of
longint;
begin
readln(n);
for i:=1 to n do
for j:=1 to i do
  a[i,j]:=j*trunc(exp((n-i)*ln(3)));
for i:=1 to n do
begin
for j:=1 to n do
write(a[i,j], ' ');
writeln;
end; end.

```

SUBIECTUL III

Se propune modificarea textului problemei de la sub. 3:

Subprogramul cifra primește prin singurul său parametru x , un număr real nenul pozitiv și **întoarce valoarea cifrei unităților** părții întregi a lui x . (În loc de furnizează prin parametrul y valoarea cifrei unităților, deoarece în exemplul de apel a funcției cifra, nu apare parametrul y !)

1. $\frac{1}{2}$
 2. $\frac{1}{7}$ pentru intensiv, respectiv 456 pentru neintensiv
 3.

a) Se face conversia spre întreg a parametrului real x . Funcția **cifra** întoarce ultima cifra a numărului întreg astfel obținut.

Varianta C/C++

```

int cifra(float x)
{return (int)x % 10;
}

```

b)

Varianta C/C++

```

#include <fstream.h>
int n;
float mediasem1, mediasem2;
int cifra(float x)
{return (int)x % 10;
}
void main()
{fstream f("medii.txt",ios::in);
int c1,c2;
while(f>>mediasem1>>mediasem2)
{
218

```

Varianta PASCAL

```

function cifra(x:real):integer;
begin
x:=x*100;
cifra:=round(x) div 100;
end;

```

Varianta PASCAL

```

program varianta60;
var f:text;
    n,c1,c2:integer;
    mediasem1,mediasem2:real;
function cifra(x:real):integer;
begin
x:=x*100;
cifra:=round(x) div 100;
end;
begin
assign(f,'medii.txt');

```

```

c1= cifra(mediasem1);
c2=cifra(mediasem2);
if(c1<c2) n++;
}
cout<<n<<endl;
f.close();
}

```

```

reset(f);
n:=0;
while not eof(f) do
begin
  readln(f,mediasem1,mediasem2);
  c1:=cifra(mediasem1);
  c2:=cifra(mediasem2);
  if(c1<c2) then n:=n+1;
end;
writeln(n);
end

```

c) O posibilă rezolvare a problemei ar fi parcurgerea secvențială a fișierului (până se atinge sfârșitul de fișier), citirea tuturor numerelor reale scrise în fișier și reținerea lor într-un vector. Se parcurge apoi acest vector (secvențial, câte două elemente deodată) se verifică proprietatea din enunțul problemei: ultima cifra a părții întregi a primului număr din pereche să fie strict mai mică decât ultima cifră a părții întregi a celui de al doilea număr din pereche.

O altă metodă mai eficientă atât din punct de vedere al spațiului de memorie ar fi cât și al timpului de execuție ar fi citirea din fișier a câte unei perechi de numere reale (mediasem1 și mediasem2), compararea ultimei cifre a părții întregi a primului număr cu ultima cifră a părții întregi a celui de al doilea număr citit și contorizarea, după caz, a soluțiilor problemei.

- Prima metodă necesită utilizarea unui vector cu cel mult 1200 elemente reale iar a doua metodă nu folosește vectorul;
- La prima metodă se parcurg numerele citite de două ori: odată la citirea din fișier și apoi la parcurgerea vectorului pentru rezolvarea problemei. A doua metodă, cea descrisă de algoritmul, nu mai necesită parcurgerea vectorului, comparația realizându-se la citirea din fișier a perechii curente de numere reale.

Varianta <62>

SUBIECTUL I

Problema 2: Eroare de tipar !!! « Scrieți valorile care se afișează..... »

1. b

2.

a) 8162 ; 2816 ; 6281 ; 1628

b) Oricare dintre numerele 1000 ; 2000 ; ... (orice număr de forma $c \cdot 10^n$, unde c este o cifră nenulă iar n este un număr natural, $n \geq 3$)

d)

citește x (număr natural)

```

aux ← x
-- repetă
  c ← x%10
  x ← [x/10]
  t ← x
  -- dacă c=0 atunci
    aux ← x
  -- repetă

```

```

-- dacă t ≠ 0 atunci
    c ← c*10
    t ← [t/10]
-- ■
-- ■ până când t=0
    c ← c+x
    scrie x
-- ■ până când x=aux și c ≠ 0
    
```

c)

Varianta C/C++

```

#include <iostream.h>
void main()
{unsigned long int x,t,aux,c;

    cin>>x;
    do
    { c=x%10;
      x/=10;
      t=x;
      if(c==0)
        aux=x;
      while(t!=0)
        { c*=10;
          t/=10;
        }
      x+=c;
      cout<<x;
    } while(x!=aux || c==0);
    
```

Varianta PASCAL

```

program varianta62;
var x,aux,t,c:longint;
begin
    readln(x);
    aux:=x;
    repeat
        c:=x mod 10;
        x:= x div 10;
        t:=x;
        if(c=0) then
            aux:=x;
        while (t<>0) do
            begin
                c:=c*10;
                t:=t div 10;
            end;
        x:=c+x;
        writeln (x)
    until (x=aux) and (c<>0);
end.
    
```

SUBIECTUL II

1. a

2. b

3. Pentru intensiv:

0	1	1	0	0	1
0	1	0	0	0	0
0	0	0	0	0	0
0	0	1	0	0	0
0	0	1	0	0	0
0	0	1	0	0	0

 Pentru neintensiv : 3

0	1	1	0	0	1
0	1	0	0	0	0
0	0	0	0	0	0
0	0	1	0	0	0
0	0	1	0	0	0
0	0	1	0	0	0

4. e

5. Explicații

Se citește de la tastatură șirul de caractere **sir**. Programul creează șirul de caractere **litere**, format din literele mici ale alfabetului englez care apar în șirul citit de la tastatură, prin parcurgerea secvențială a șirului de caractere

Se citește de la tastatură șirul de caractere **sir1**. Programul creează șirul de caractere **sir2**, format din literele mici ale alfabetului englez care apar în șirul citit de la tastatură, prin parcurgerea secvențială a șirului de caractere **sir1**. Se utilizează două variabile de lucru de tip mulțime, cu tipul de bază caracter : variabila **litere**, care reține toate literele mici ale alfabetului englez și variabila **litere_distincte** care

sir. Dacă elementul curent (**i**) al lui **sir** este literă mică și nu aparține șirului **litere** creat (`!strchr(litere,sir[i])`), se adaugă la șirul **litere** un spațiu și litera găsită.

Varianta C/C++

```
#include <iostream.h>
#include <string.h>
int i,apartine,l,j;
char sir[256],litere[255];
void main()
{cin.get(sir,256);
for(i=0;i<strlen(sir);i++)
    if(sir[i]>='a' &&
sir[i]<='z' &&
!strchr(litere,sir[i]))
    { strcat(litere," ");

litere[strlen(litere)-
2]=sir[i];
    }
cout<<litere;
}
```

reține literele distincte ale șirului format, **sir2**. Dacă elementul curent (**i**) al lui **sir** este literă mică (aparține mulțimii **litere**) și nu aparține șirului **sir2** creat (nu aparține mulțimii **litere_distincte**), se adaugă la șirul **sir2** un spațiu și litera găsită.

Varianta PASCAL

```
program varianta62;
var sir1,sir2:string;
    i:integer;
    litere,litere_distincte:set of char;
begin
readln(sir1);
sir2:=''; litere_distincte:=[]; litere:=[];
for i:=ord('a') to ord('z') do
    litere:=litere+[char(i)];
for i:=1 to length(sir1) do
    if (sir1[i] in litere) and
not(sir1[i] in litere_distincte)
then
begin
sir2:=sir2+sir1[i]+' ';

litere_distincte:=litere_distincte+[sir1[i]];
end;
writeln(sir2);
end
```

SUBIECTUL III

1. a

2. 9 pentru **intensiv**, respectiv 26944962 pentru **neintensiv**

3.

Șirul lui Fibonacci este o secvență de numere în care fiecare număr se obține din suma precedentelor două din șir. Astfel, primele 10 numere ale șirului lui Fibonacci sunt:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...

(primele 2 numere sunt predefinite, iar restul se obțin în mod recursiv, din suma precedentelor două: $3 = 2 + 1$, $5 = 3 + 2$, șamd...)

Varianta C/C++

```
#include <iostream.h>

long int a,b,c,n=1,nr_pe_linie=1;
void main()
{a=0;b=1;
cout<<1<<' ';
while(n<40)
{c=a+b;
a=b;b=c;
if(c%2==1)
{cout<<c<<' ';
n++; nr_pe_linie++;
if(nr_pe_linie==5)
{ cout<<endl;nr_pe_linie=0;}
} } }
```

Varianta PASCAL

```
Program varianta62;
var a,b,c:longint;
n,nr_pe_linie:integer;
begin
n:=1;nr_pe_linie:=1;
write(1,' ');
a:=0;b:=1;
while(n<40) do
begin
c:=a+b;a:=b;b:=c;
if(c mod 2=1) then
begin
write(c,' ');

n:=n+1;nr_pe_linie:=nr_pe_linie+1;
if(nr_pe_linie=5) then
begin
nr_pe_linie:=0;writeln;
```

4. a)

Varianta C/C++

```
#include <fstream.h>
#include <values.h>
int a[10000],b[10000],x,na,nb;
void adaug_pare(int x)
{int stg=1,dr=na,m,gasit=0,i;
if(na==0) {a[1]=x;na=1;}
else
do{ m=(stg+dr)/2;
  if(x==a[m]) gasit=1;
  else if(x>a[m])
    if(x<a[m-1])
      {for(i=na+1;i>m;i--)
        a[i]=a[i-1];
        a[m]=x; gasit=1;na++;
      }
    else
      {dr=m-1;
m=(stg+dr)/2; }
    else
      if(x>a[m+1])
        {for(i=na+1;i>m+1;i-- )
a[i]=a[i-1];
        a[m+1]=x; gasit=1;
na++; }
    else
      {stg=m+1;
m=(stg+dr)/2; }
  } while(gasit==0);
}
void adaug_impere(int x)
{int stg=1,dr=nb,m,gasit=0,i;
if(nb==0) {b[1]=x;nb=1;}
else
do{ m=(stg+dr)/2;
  if(x==b[m]) gasit=1;
  else if(x<b[m])
    if(x>b[m-1])
      {for(i=nb+1;i>m;i--)
b[i]=b[i-1];
        b[m]=x;
gasit=1;nb++; }
    else
      {dr=m-1;
m=(stg+dr)/2; }
    else
      if(x<b[m+1]) {
for(i=nb+1;i>m+1;i--) b[i]=b[i-1];
        b[m+1]=x; gasit=1;
nb++; }
    else
```

```
end;
end;
end;
end.
```

Varianta PASCAL

```
Program Varianta62;
var a,b:array[0..10000] of integer;
x,na,nb,i:integer;
f,g:text;
procedure adaug_pare(x:integer);
var stg,dr,m,gasit,i:integer;
begin
stg:=1;dr:=na;gasit:=0;
if(na=0) then
begin a[1]:=x;na:=1;end
else
repeat
m:=(stg+dr) div 2;
if(x=a[m]) then gasit:=1
else
if(x>a[m]) then
if(x<a[m-1]) then
begin
for i:=na+1
downto m+1 do
a[i]:=a[i-1];
a[m]:=x;
gasit:=1;na:=na+1;
end
else
begin
dr:=m-1;
m:=(stg+dr) div 2;
end
else
if(x>a[m+1]) then
begin
for i:=na+1
downto m+1 do
a[i]:=a[i-1];
a[m+1]:=x; gasit:=1;
na:=na+1;
end
else
begin
stg:=m+1;
m:=(stg+dr) div 2;
end
until gasit=1;
end;
procedure adaug_impere( x:integer);
var stg,dr,m,gasit,i:integer;
begin
stg:=1;dr:=nb;gasit:=0;
if(nb=0) then begin b[1]:=x;nb:=1;end
else
```

```

        {stg=m+1;
m=(stg+dr)/2; }
        } while(gasit==0);
}
void main()
{fstream f("date.in",ios::in);
a[0]=MAXINT;
while(f>>x)
    if(x%2==0) adaug_pare(x);
    else adaug_impere(x);
f.close();
fstream g("date.out",ios::out);
for(int i=1;i<=nb;i++)
    g<<b[i]<<' ';
g<<endl;
for(i=1;i<=na;i++)
    g<<a[i]<<' ';
g<<endl;
g.close();

```

```

repeat
    m:=(stg+dr) div 2;
    if(x=b[m]) then gasit:=1
    else
        if(x<b[m]) then
            if(x>b[m-1]) then
                begin
                    for i:=nb+1
downto m+1 do
                        b[i]:=b[i-1];
                        b[m]:=x;
gasit:=1;nb:=nb+1;
                    end
                else
                    begin dr:=m-1;
m:=(stg+dr) div 2;
                    end
                else
                    if(x<b[m+1]) then
                        begin
                            for i:=nb+1
downto m+2 do
                                b[i]:=b[i-1];
                                b[m+1]:=x; gasit:=1;
nb:=nb+1;
                            end
                        else
                            begin
                                stg:=m+1;
m:=(stg+dr) div 2;
                            end
                        until gasit=1;
                    end;
begin
assign(f,'date.in'); reset(f);
assign(g,'date.out'); rewrite(g);
a[0]:=MAXINT;
while( NOT eof(f)) do
begin
    read(f,x);
    if(x mod 2=0) then adaug_pare(x)
    else adaug_impere(x);
end;
close(f);
for i:=1 to nb do
    write(g,b[i],' ');
writeln(g,' ');
for i:=1 to na do
    write(g,a[i],' ');
close(g);
end.

```

b) Valorile citite din fișier se adaugă în doi vectori : vectorul **a**, ordonat descrescător, format din numerele pare și vectorul **b**, ordonat descrescător, format din numerele impare. Se utilizează metoda sortării prin inserție: se caută poziția noului element în vector (poziția

m) (folosind metoda căutării binare). Dacă elementul nu se regăsește în vector, atunci se deplasează spre dreapta toate elementele vectorului de index mai mare decât **m** iar elementul nou se așează pe poziția **m**. În acest mod, se vor reține doar elementele distincte.

SUBIECTUL I

1. a

2.

b) 19, 319, 21 (Oricare trei numere naturale distincte impare)

c) citește n, d (numere naturale) $b \leftarrow 0$ $v \leftarrow 0$

```

-- pentru i ← 1, n execută
  citește x ( număr natural nenul)
  a ← 0
  aux ← x
  -- repetă
  -- dacă x % d = 0 atunci
    a ← a + 1
    x ← [x/d]
  până când x % d ≠ 0
  -- dacă a > b atunci
    b ← a
    v ← aux
scrie v, ' ', b

```

d)

Varianta C/C++

```

#include <iostream.h>
void main()
{unsigned long int n,d,b,v,x,aux,a;
cin>>n>>d;
b=0;
v=0;
for(int i=1;i<=n;i++)
{cin>>x;
a=0;
aux=x;
while( x%d==0)
{ a+=1;
x/=d;
}
if(a>b)
{ b=a;
v=aux;
}
}
cout<<v<<' '<<b;
}

```

Varianta PASCAL

```

Program varianta 63;
var n,d,b,v,a,aux,i,x:integer;
begin
readln(n,d);
b:=0;
v:=0;
for i:=1 to n do
begin
readln(x);
a:=0;
aux:=x;
while(x mod d=0) do
begin
a:=a+1;
x:=x div d;
end;
if(a>b) then
begin
b:=a;
v:=aux;
end;
end;
writeln(v, ' ', b);

```


end.

SUBIECTUL II

1. b

2. b

3. 3 ; 5 ; 6

4. Varianta C/C++ pentru intensiv

```
if (strcmp(s1, s1) < 0)      cout << s1 << '\n' << s2;
else cout << s2 << '\n' << s1;
```

Varianta C/C++ pentru neintensiv

```
if (strlen(s1) < strlen(s2))
    cout << s1 << s2;
else cout << s2 << s1;
```

5. Explicații

Se observă că elementul aflat pe linia i și coloana j a matricei se calculează după formula :

$a[i][j] = n * (j - 1) + i$, unde n este numărul de linii ale matricei.

Varianta C/C++

```
#include <iostream.h>
int a[51][51], m, n, i, j;
void main()
{ cin >> n >> m;
  for (j = 1; j <= m; j++)
    for (i = 1; i <= n; i++)
      a[i][j] = (j - 1) * n + i;
  for (i = 1; i <= n; i++)
    { for (j = 1; j <= m; j++)
      cout << a[i][j] << ' ';
    }
}
```

Varianta PASCAL pentru intensiv

```
If (strcmp(s1, s2) < 0)      writeln(s1, '\n', s2)
Else writeln(s2, '\n', s1);
```

Varianta PASCAL pentru neintensiv

```
if length(s1) < length(s2) then
    writeln(s1, s2)
else writeln(s2, s1);
```

Varianta PASCAL

```
program varianta63;
var a: array[1..50, 1..50] of integer;
    n, m, i, j: integer;
begin
  readln(n, m);
  for j := 1 to m do
    for i := 1 to n do
      a[i, j] := n * (j - 1) + i;
  for i := 1 to n do
    begin
      for j := 1 to m do
        write(a[i, j], ' ');
      writeln;
    end;
end.
```

SUBIECTUL III

1. c

2. 15 pentru intensiv, respectiv 6 pentru neintensiv

3.

Varianta C/C++

```
int verific(long int x)
{ while (x >= 100)
  { if (x % 2 == 1 && (x / 10) % 2 &&
    (x / 100) % 2)
    return 1;
    x /= 10;
  }
  return 0;
}
```

Varianta PASCAL

3. a)

```
function
verif(x: longint): boolean;
begin
  c, y: integer;
  while (x > 0) do
    begin
      c := x mod 10;
      if (c mod 2 = 0) then
        else
          y = y + 1;

      if (y = 3) then
        verific = 1;
      x := x div 10;
    end;
end;
```

```

    verif=0;
end;

3. b)
program pct3;
var
    n,i:integer;
    t:Boolean;
    x:longint;

function
verif(x:longint):boolean;
begin
    c,y:integer;
    while(x>0) do
    begin
        c:=x mod 10;
        if (c mod 2=0) then
        else
            y=y+1;
            if (y=3) then
                verif=1;
            x:=x div 10;
        end;
        verif=0;
    end;

begin
    assign(f,'date.txt');
    reset(f);
    readln(f,n);
    t:=true;
    for i:=1 to n do
    begin
        read(f,x);
        if (verif(x)) then
            begin
                write('x=',x,' ');
                t:=false;
            end ;
        end ;
        if (t) then
            write('NU') ;
    end.

```

3.c)

Eficiența algoritmului constă în faptul că se folosește o memorie minimă, iar operațiile sunt simplificate.

- se citesc cele n numere, fără a se stoca în memorie și se prelucrează la citire
- subprogramul **verif()** va returna 1 la prima secvență de 3 cifre găsită, fără să le gasească pe toate.

Astfel, se economiseste timp.

b) Varianta C/C++

```
#include <fstream.h>
long int x,n,ok;
int verific(long int x)
{while(x>=100)
  {if(x%2==1 && (x/10)%2 &&
(x/100)%2 )
    return 1;
  x/=10;
  }
return 0;
}
void main()
{fstream f("date.txt",ios::in);
f>>n;
for(int i=1;i<=n;i++)
  {f>>x;
  if( verific(x/1000) )
    {ok=1; cout<<x<<' '; }
  }
if(!ok) cout<<"nu";
}
```

Varianta PASCAL

```
program varianta63;
var f:text;
    n,i,exista:integer;
    x:longint;
function verific(x:longint):integer;
var ok:integer;
begin
ok:=0;
while(x>=100) and (ok=0) do
  if(x mod 2=1) and ((x div 10) mod
2=1) and ((x div 100) mod 2=1) then
    ok:=1
  else x:=x div 100;
verif:=ok;
end;
begin
assign(f,'date.txt');reset(f);
readln(f,n); exista:=0;
for i:=1 to n do
begin
  readln(f,x);
  if (verif(x div 1000)=1) then
  begin
    write(x,' '); exista:=1;
  end;
end;
if exista=0 then writeln('nu');
close(f);
end.
```

Funcția **verif()** are ca parametru de intrare un număr natural . Se testează succesiv paritatea ultimelor trei cifre ale parametrului de intrare. In programul principal, se parcurg succesiv cele **n** numere naturale citite din fișierul text. Se testează doar paritatea primelor trei cifre ale fiecărui număr **x** citit din fișier, prin apelul funcției **verif()** cu parametru de intrare **[x/1000]**.

Varianta <64>

SUBIECTUL I

1. c

2.

a) Vor fi afișate, în ordine, valorile: 2, 3, 4, 4, 5, 6, 5, 6, 7, 8, 10

b) n=15

c) Varianta C/C++

```
#include<iostream.h>
unsigned n,k,i,j;

void main(){
cin>>n;
for(i=1;i<=n;i++)
  for(j=1;j<=n;j++){cout<<i+j<<' ';
```

Varianta PASCAL

```
Program var_64_I;
var n,k,i,j:word;

begin
readln(n);
for i:=1 to n do
  for j:=1 to n do begin
```

```
k++;}
cout<<k<<endl;}
```

```
write(i+j,' ');k:=k+1; end;
writeln(k) end.
```

d) citeste n

$k \leftarrow 0$

$i \leftarrow 1$

Cat timp $i \leq n$ executa

| $j \leftarrow 1$
| Cat timp $j \leq i$ executa

| | Scrie $i+j$

| | $k \leftarrow k+1$

| | $j \leftarrow j+1$

| $i \leftarrow i+1$

Scrie k

SUBIECTUL II

1. intensiv

d

2. intensiv

a

3. 3

4. 3

5. Explicații

Se parcurge șirul și la întâlnirea caracterului spațiu sau la sfârșitul șirului se verifică dacă litera precedentă este vocală și prima literă a cuvântului (caracterul de pe poziția k) este, de

asemenea, vocală.

Varianta C/C++

```
#include<iostream.h>
#include<string.h>
char x[200];
int n,k,t,i;
void main()
{
cin.get(x,200);
char y[10]="aeiouAEIOU";
k=0;
for(t=0;t<=strlen(x);t++)
if(x[t]!=' '||t==strlen(x))
if(strchr(y,x[k])&&strchr(y,x[t-1]))
{n++; k=t+1;}
if(!n) cout<<"Nu exista";
else cout<<n;
}
```

neintensiv

d

neintensiv

a

Varianta PASCAL

```
Program var_64_II;
var x:string[200];
n,k,t,i:integer;
m:set of char;
begin
m=['a','e','i','o','u','A','E','I','O','U'];
readln(x);
x:=x+' ';
k:=1;
for t:=1 to t<=length(x) do
if (x[t]=' ') then
if(x[k] in m)and (x[t-1] in m) then
begin
inc(n); k=t+1; end;
if n=0 then write('Nu exista')
else write(n) end.
```

SUBIECTUL III

1. intensiv

d

2. intensiv

$5 \cdot 10 = 50$

$5 \cdot 9 = 45$

neintensiv

d

neintensiv

$9 \cdot -1 = -9$

$9 \cdot 0 = 0$

5*8 = 40
 5*7 = 35
 5*6 = 30
 5*5 = 25
 5*4 = 20
 5*3 = 15
 5*2 = 10
 5*1 = 5
 5*0 = 0

3. Varianta C/C++

```
int diviz(int n, int a[])
{int i, j, k=0;
for(i=1; i<n; i++)
    for(j=i+1; j<=n; j++)

if(a[i]%a[j]==0 || a[j]%a[i]==0) k++;
return k;
}
```

4. a)

Varianta C/C++ intensiv

```
#include<iostream.h>
#include<fstream.h>
ifstream f("date.in");

long x[1000], n, i, j, max, k;
void main()
{while(!f.eof())
f>>k;
if(!f.eof()){n++;x[n]=k;}
i=1;
while(i<n)
{j=i+1;
while(j<=n&& x[j-1]>x[j]) j++;
if(j-i>max){max=j-i;k=i;}
i=j;
}
for(i=k; i<=k+max-1; i++) cout<<x[i]<<'
\';
cout<<endl;}
```

Varianta C/C++ neintensiv

```
#include<iostream.h>
#include<fstream.h>
ifstream f("date.in");
long x[1000], n, i, j, max, k;
void main()
{while(!f.eof())
f>>k;
```

9*1=9
 9*2=18
 9*3=27
 9*4=36
 9*5=45
 9*7=63
 9*8=72
 9*9=81

Varianta PASCAL

```
Type vector=array[1..200] of
integer;
Function diviz(n:integer;
a:vector):integer;
var i, j, k:integer;
begin
k:=0;
for i:=1 to n-1 do
for j:=i+1 to n do
    if(a[i] mod a[j]=0)or(a[j] mod
a[i]=0) then inc(k);
diviz:=k;
end;
```

Varianta PASCAL intensiv

```
Program var_64_III;
var x:array[1..1000] of longint;
k, n, i, j, max:integer;

f:text;
begin
assign(f, 'date.in'); reset(f);
while not eof(f) do
begin
read(f, k); inc(n); x[n]:=k;
end;
i:=1;
while i<n do begin
j:=i+1;
while(j<=n)and(x[j-1]>x[j]) do
j:=j+1;
if j-i>max then begin max:=j-i;
k:=i
end;
i:=j; end;
for i:=k to k+max-1 do
write(x[i], ' ');
writeln end.
```

Varianta PASCAL neintensiv

```
Program var_64_III;
var x:array[1..1000] of longint;
k, n, i, j, max:integer;
f:text;
begin
assign(f, 'date.in'); reset(f);
while not eof(f) do
```

```

if(!f.eof()){n++;x[n]=k;}
i=1;
while(i<n)
{j=i+1;
while(j<=n&&x[j-1]>x[j]) j++;
if(j-i>max)max=j-i;
i=j;
}
cout<<max<<endl;}

```

```

begin
read(f,k); inc(n);x[n]:=k;
end;
i:=1;
while i<n do begin
j:=i+1;
while(j<=n)and(x[j-1]>x[j]) do
j:=j+1;
if j-i>max then max:=j-i;
i:=j; end;
writeln(max) end.

```

b) **intensiv**

Se determină secvența descrescătoare de lungime maximă printr-o singură parcurgere a șirului, reținând în variabila k poziția de început a secvenței de lungime maximă. După generarea unei secvențe între indicii i și $j-1$, următoarea secvență se va căuta de la poziția j .

neintensiv

Se determină secvența descrescătoare de lungime maximă printr-o singură parcurgere a șirului astfel încât după generarea unei secvențe între indicii i și $j-1$, următoarea secvență se va căuta de la poziția j .

Varianta <65>

SUBIECTUL I – intensiv informatică

1. b

2. a) Vor fi afișate valorile: 13277231 și 1

b) Soluție este orice număr în care cifra unităților apare de 4 ori. De exemplu: 1111 și

20222

Varianta C/C++

```

#include<iostream.h>
unsigned n,m,c,v,u;
void main(){
cin>>n;
m=0; v=n;
u=n%10;
do{
c=n%10; v=v*10+c;
if(c==u) m++;
n=n/10;
}
while(n);
cout<<v<<' '<<m<<endl;}

```

Varianta PASCAL

```

Program var_65_I;
var n,m,c,v,u:word;
begin
readln(n);
m:=0; v:=n;
u:=n mod 10;
repeat
c:=n mod 10;
v:=v*10+c;
if c=u then inc(m);
n:=n div 10;
until n=0;
writeln(v,' ',m);
end.

```

d) citeste n

m←0; v←n

u←n%10

Cat timp n≠0 executa

| c← n%10

| v←v*10+c

| daca c=u atunci

| | m←m+1

| | ■

| ■ n←[n/10]

Scris v,m

SUBIECTUL II

1. c

2. d

3. 2 muchii

4. 101 noduri

5. Explicații

Se observă că liniile impare se completează cu valori de la primul element la ultimul, iar cele pare de la ultimul la primul. Se folosește variabila k pentru a genera elementele care se vor completa pe linii, valoarea lui k mărindu-se cu 1 la fiecare pas.

Varianta C/C++

```
#include<iostream.h>
int n,m,i,j,k,a[51][51];
void main()
{ cin>>n>>m; k=1;
for(i=1;i<=n;i++)
if(i%2)
    for(j=1;j<=m;j++)a[i][j]=k++;
else
    for(j=m;j>=1;j--)a[i][j]=k++;
for(i=1;i<=n;i++)
    {for(j=1;j<=m;j++)
        cout<<a[i][j]<<' \';
        cout<<endl;
    }
}
```

Varianta PASCAL

```
Program var_65_II;
var n,m,i,j,k:integer;
    a:array[1..50,1..50] of
integer;
begin
readln(n,m);
k:=1;
for i:=1 to n do
    if i mod 2 <>0 then
        for j:=1 to m do
            begin a[i,j]:=k;
inc(k);
end
        else
            for j:=m downto 1 do
                begin a[i,j]:=k;
inc(k);
end;
for i:=1 to n do
    begin
        for j:=1 to m do write(a[i,j],
');
writeln end end.
```

SUBIECTUL III

1. a

2.222

3. a) Varianta C/C++

```
int sfx(long x)
{
if(x%10<x/10%10&& x/10%10< x/100%10)
return 1;
else return 0;
}
```

b)

Varianta C/C++

```
#include<iostream.h>
#include<fstream.h>
ifstream f("date.in");
long x,n;
int sfx(long x)
{
if(x%10<x/10%10&& x/10%10< x/100%10)
return 1;
else return 0;
}
```

Varianta PASCAL

```
function sfx(x:longint):integer;
begin
if (x mod 10<x div 10 mod 10)and(x
div 10 mod 10<x div 100 mod 10)then
sfx:=1
else sfx:=0 end;
```

Varianta PASCAL

```
Program var_64_III;
var x,n:longint;
f:text;
function sfx(x:longint):integer;
begin
if (x mod 10<x div 10 mod 10)and(x
div 10 mod 10<x div 100 mod 10)then
sfx:=1
else sfx:=0 end;
```

```

}
void main()
{
while(!f.eof())
{
f>>x;
if (!f.eof())
    if (sfx(x)&&sfx(x/10)&&sfx(x/100)&&
sfx(x/1000)) n++;}

cout<<n<<endl;
}

```

```

begin
assign(f,'date.in'); reset(f);
while not eof(f) do
begin
read(f,x);
if (sfx(x)=1)and(sfx(x div
10)=1)and
(sfx(x div 100)=1)and(sfx(x div
1000)=1) then inc(n);end;
writeln(n);
close(f);
end.

```

b) Pentru a utiliza eficient memoria, nu stocăm valorile din fișier într-un tablou, ci verificăm fiecare element imediat după preluarea lui din fișier.

Deoarece fiecare număr citit are **exact 6 cifre**, folosim funcția **sfx** pentru a verifica, pe rând,

valorile: x , $[x/10]$, $[x/100]$ și $[x/1000]$.

Varianta <66>

SUBIECTUL I

1. b

2.

a) Se va afișa NU

b) 10, 20, 1, 4, 0

c) Varianta C/C++

```

#include<iostream.h>
int n;
unsigned x;
void main(){
n=0;
do{
cin>>x;
if (x) if(!(x%5)) n++; else n--;
} while (x);
if(!n) cout<<"DA"; else cout<<"NU";
}

```

d)

```

n←0
citeste x
Cat timp x≠0 executa
|   daca x%5=0 atunci n←n+1
|   |   altfel n←n-1
|   |   |
|   |   |   citeste x
|   |   |
|   |   |
Daca n=0 atunci scrie "DA"
|   altfel scrie "NU"
|

```

Varianta PASCAL

```

Program var_66_I;
var n:integer;
    x:word;
begin
n:=0;
repeat
readln(x);
if x<>0 then
    if x mod 5=0 then n:=n+1

else n:=n-1;
until x=0;
if n=0 then writeln('DA')
    else writeln('NU')

end.

```

SUBIECTUL II

1. c

2. intensiv

b

3. 3 și 4

4. intensiv

Secvența afișează numărul întreg din ultimul nod al listei.

5.

Varianta C/C++

```
#include<iostream.h>
#include<string.h>
char x[50];
int i,j;
void main()
{
cin>>x;
for(i=0;i<strlen(x);i++)
{
for(j=0;j<=i;j++)
cout<<x[j];
cout<<endl;
}
}
```

SUBIECTUL III

1. intensiv

c

2. intensiv

Funcția are valoarea 48.

3. Varianta C/C++

```
long calcul(unsigned n, long a[])
{long d,x=a[1],y=a[2];
if(!x&&!y)d=0;
else
if(x&&!y) d=x;
else
if(!x&&y) d=y;

else
{while(x!=y)if(x>y) x-=y;
else y-=x;
d=x;}
for(int i=3;i<=n;i++)
{
x=d; y=a[i];
if(!x&&!y)d=0;
else
if(x&&!y) d=x;
else
if(!x&&y) d=y;
else
{while(x!=y)if(x>y) x-=y;
else y-=x;
d=x;}
}
}
```

neintensiv

b

Neintensiv

1 1 1 1 1
4 4 4 4 4
3 3 3 3 3
2 2 2 2 2
5 5 5 5 5

Varianta PASCAL

```
Program var_66_II;
var x:string[50];
i:integer;
begin
readln(x);
for i:=1 to length(x) do
writeln(copy(x,1,i));
end.
```

neintensiv

c

neintensiv

Lipsește secvența 222.

Varianta PASCAL

```
Type vector=array[1..10000] of
longint;
Function calcul(n:word;
a:vector):longint;
var d,x,y:longint;
begin
x:=a[1];y:=a[2];
if(x=0)and(y=0) then d:=0
else
if(x<>0)and(y=0) then d:=x
else
if(x=0)and(y<>0) then d:=y
else
begin
while x<>y do
if x>y then x:=x-y
else y:=y-x;
d:=x;
end;
for i:=3 to n do
begin
x:=d; y:=a[i];
if(x=0)and(y=0) then d:=0
else
```

```
return d;
}
```

```
if(x<>0)and(y=0) then d:=x
else
if(x=0)and(y<>0) then d:=y
else
begin
while x<>y do
if x>y then x:=x-y
else y:=y-x;
d:=x;
end;
end;
calcul:=d;
end;
```

4. a)

Varianta C/C++

```
#include<iostream.h>
#include<fstream.h>
ifstream f("a.txt");
ifstream g("b.txt");
long x, min;
int k;
void main()
{
g>>x; min=x;
while(!g.eof())
{
g>>x;
if(x<min)min=x;
}
While(!f.eof())
{
f>>x;
if(!f.eof())if(x<min)k++;
}
cout<<k;
}
```

Varianta PASCAL

```
Program var_66_III;
var x, min:longint;
k:integer;
f,g:text;
begin
assign(f,'a.txt'); reset(f);
assign(g,'b.txt'); reset(g);
readln(g,x); min:=x;
while not eof(g) do
begin
readln(g,x);
if x<min then min:=x;
end;
while not eof(f) do
begin
readln(f,x);
if x<min then k:=k+1;
end;
writeln(k);
end.
```

b)

Se determină cea mai mică valoare dintre numerele citite din fișierul **b.txt**, iar apoi se determină numărul de numere preluate din fișierul **a.txt** mai mici decât valoarea minimă determinată.

Varianta <67>

SUBIECTUL I

1. c

2.

a) 264

b) 687

c) Varianta C/C++

```
#include<iostream.h>
unsigned n,z,c;

void main()
{
cin>>n; z=0;
while(n)
{
```

Varianta PASCAL

```
Program var_67_I;
var n,z,c:word;

begin
readln(n); z:=0;
while n<>0 then
begin
```

```
c=n%10;
n=n/10;
if(c<5) z=z*10+2*c;
}
cout<<z;
}
```

d) citește n

z←0

Repetă

```
|   c←n%10
|   n←[n/10]
|   dacă c<5 atunci
|       |   z←z*10+2*c
|       |   █
```

| până când n=0

Scrie z

```
c:=n mod 10;
n:=n div 10;
if c<5 then z:=z*10+2*c;
end;
writeln(z);
end.
```

SUBIECTUL II

1.

a

2. intensiv

C

3. 2

4. intensiv

222

5. Varianta C/C++

```
#include<iostream.h>
#include<string.h>
char x[50];
int i,j;
void main()
{
cin>>x;
for(i=strlen(x)-1;i>=0;i--)
{
for(j=i;j<strlen(x);j++)
cout<<x[j];
cout<<endl;
}}
```

neintensiv

c

neintensiv

1 1 1 1

1 2 2 2

1 2 3 3

1 2 3 4

Varianta PASCAL

```
Program var_67_II;
var x:string[50];
i:integer;
begin
readln(x);
for i:=length(x) downto 1 do
writeln(copy(x,i,length(x)));
end.
```

SUBIECTUL III

1. intensiv

a

2. intensiv

- Pentru u=4 alfa(u) va avea valoarea 25.

- alfa(6) va avea valoarea 54

3. Varianta C/C++

```
int calcul(int n, long a[])
{int i,k=0,j,t;
for(i=1;i<=n;i++)
```

neintensiv

a

neintensiv

Lipsește secvența 422

Varianta PASCAL

```
Type vector=array[1..10000] of
longint;
Function calcul(n:integer;
```

```

    if(a[i]>=2)
    {
        t=1;
        for(j=2;j<=a[i]/2&&t;j++)
            if(a[i]%j==0) t=0;
        if(t) k++;
    }
    return k;
}

```

4. a)

Varianta C/C++

```

#include<iostream.h>
#include<fstream.h>
ifstream f("numere.txt");
long x,n,i,j,y;
void main()
{
    f>>n;
    i=1; f>>x;
    while(i<=n)
    {
        cout<<x<<' \';
        j=i+1; f>>y;
        while(j<=n&&y==x)
            {j++;f>>y;}
        x=y; i=j;
    }
}

```

```

a:vector):integer;
var i,j,k:integer;
    t:boolean;
begin
    k:=0;
    for i:=1 to n do
        if a[i]>=2 then begin
            t:=true;
            for j:=2 to a[i] div 2 do
                if a[i] mod j=0 then
                    t:=false;
            if t then inc(k); end;
        calcul:=k;
    end;

```

Varianta PASCAL

```

Program var_67_III;
var x,n,i,j,y:longint;
    f:text;
begin
    assign(f,'numere.txt'); reset(f);
    readln(f,n);
    i:=1; read(f,x);
    while i<=n do
        begin
            write(x,' ');
            j:=i+1; read(f,y);
            while (j<=n)and(y=x) do
                begin
                    inc(j);
                    read(f,y);
                end;
            x:=y; i:=j;
        end;
    end.

```

b) Se folosește faptul că șirul este ordonat crescător. Nu stocăm numerele citite într-un vector, ci vom folosi faptul că numerele sunt în ordine crescătoare, deci cele egale sunt dispuse pe poziții consecutive în fișier. Un element care se repetă va fi afișat pe ecran o singură dată, dublurile lui fiind doar citite din fișier, nu și afișate pe ecran.

Varianta <68>

SUBIECTUL I

1. a

2.

a) 2

b) 7, 15, 4, 18, 12, 3

c) **Varianta C/C++**

```

#include<iostream.h>
unsigned x,nr;
int n;
void main(){
    cin>>x;
    nr=0;

```

Varianta PASCAL

```

Program var_68_I;
var nr,x:word;
    n:integer;
begin
    readln(x); nr:=0;
    for i:=1 to 5 do

```

```
for(i=1;i<=5;i++)
{
    cin>>n;
    if(n%x ==0) nr++;
}
cout<<nr;
}
```

d) citeste n
nr←0
i←1
Cat timp i≤5 executa
| citeste n
| daca n%x =0 atunci
| | nr←nr+1
| | ■
| | i←i+1
Scrie nr

```
begin
    readln(n);
    if n mod x =0 then
inc(nr);
    end;
writeln(nr);
end.
```

SUBIECTUL II

1.

d

2. intensiv

a

3. t: (0, 1, 1, 2, 2, 5, 5)

4. intensiv

C/C++

x->next=z;

z->next=y;

5. Varianta C/C++

```
#include<iostream.h>
int m,n,i,j,a[51][51];
long p;
void main()
{
cin>>m>>n;
for(i=1;i<=m;i++)
    for(j=1;j<=n;j++)
        cin>>a[i][j];

p=1;

for(i=2;i<=m;i+=2)
    for(j=1;j<=n;j+=2)
        p=p*a[i][j];
cout<<p<<endl;}
```

Pascal

x^.next:=z;

z^.next:=y;

neintensiv

a

neintensiv

C/C++

strlen

Pascal

length

Varianta PASCAL

```
Program var_68_II;
var m,n,i,j:integer;
    a:array[1..50,1..50] of integer;
    p:longint;
begin
readln(m,n);
for i:=1 to m do
    for j:=1 to n do begin
        readln(a[i,j]);
        if (i mod 2=0) and(j mod 2<>0)
then
            p:=p*a[i,j];
writeln(p);
end.
```

SUBIECTUL III

1. intensiv

b

2. intensiv

6 6 6 6 3

3. Varianta C/C++

```
int calcul(int n, int k)
{int p=0,c;
While(n && p<k)
```

neintensiv

b

neintensiv

Lipsește secvența 311

Varianta PASCAL

```
Function calcul(n,k:integer):integer;
var p,c:integer;
begin
```

```

    {
        c=n%10;
        p++;
        n=n/10;
    }
    if(p==k) return c;
    else return -1;
}

```

4. a)

Varianta C/C++

```

#include<iostream.h>
#include<fstream.h>
ifstream f("sir.txt");
long l,x,n,i,j,y,k;
void main()
{
    f>>n;
    l=0;
    i=1;f>>x;
    while(i<=n)
    {
        j=i+1; f>>y;
        while(j<=n&&y==x)
            {j++;f>>y;}
        if(j-i>l){l=j-i;k=x;}
        else if(j-i==l)k=x;
        x=y; i=j;
    }
    cout<<l<<' \<<k<<endl;
}

```

```

p:=0;
while (n<>0) and(p<k) do
    begin
        c:=n mod 10;
        inc(p);
        n:=n div10;
    end;
if p=k then calcul:=c
else calcul:=-1;
end;

```

Varianta PASCAL

```

Program var_68_III;
var x,n,i,j,y,l:longint;
    f:text;
begin
    assign(f,'sir.txt'); reset(f);
    readln(f,n); l:=0;
    i:=1; read(f,x);
    while i<=n do
        begin
            j:=i+1; read(f,y);
            while (j<=n)and(y=x) do
                begin
                    inc(j);
                    read(f,y);
                end;
            if j-i>l then begin l:=j-i; k:=x;
                end
            else if j-i=l then k:=x;
                x:=y; i:=j;
            end;
        writeln(l,' \',k);
    end.

```

b) Se folosește faptul că șirul este ordonat crescător. Nu stocăm numerele citite într-un vector, ci vom folosi faptul că numerele sunt în ordine crescătoare, deci cele egale sunt dispuse pe poziții consecutive în fișier. În situația în care găsim un platou de lungime mai mare modificăm valoarea variabilei care reține lungimea maximă (l) și reținem numărul corespunzător platoului (variabila k). Dacă găsim un platou de lungime maximă, modificăm valoarea variabilei k.

Varianta <69>

SUBIECTUL I

1. d

2.

a) 4789

b) x=200, y=100

c) **Varianta C/C++**

```

#include<iostream.h>
void main() {
    cin>>x>>y;
}

```

Varianta PASCAL

```

Program var_69_Word;
begin
    readln(x,y);
end;

```

```
t=0; u=1;
do
{ if (x%10>y%10) z=x%10;
  else z=y%10;
  t=t+z*u;
  u=u*10; x=x/10; y=y/10;
}
while(x||y);
cout<<t;
}
```

d) citeste x,y

t←0

u←1

Cat timp (x≠0)sau(y≠0) executa

| daca x%10 >y%10 atunci

| | z←x%10

| | altfel z ←y%10

| | ■ t ←t+z*u; u←u*10; x←[x/10]; y←[y/10];

Scrie t

```
t:=0; u:=1;
repeat
if x mod 10>y mod 10 then z:=x mod
10
                  else z:=y mod 10;
t:=t+z*u;
u:=u*10; x:=x div 10; y:=y div 10;
until (x=0)and(y=0);
writeln(t);
end.
```

SUBIECTUL II

1.

B

2. intensiv

c

3. t: (0, 1, 1, 3, 3, 4, 4)

4. intensiv

C/C++

u->next=v;

u=v;

Pascal

u^.next:=v;

u:=v;

5. Varianta C/C++

```
#include<iostream.h>
int k,n,i,j,a[51][51];
void main()
{
cin>>n;
for(i=1;i<=n;i++)
  for(j=1;j<=n;j++)
    cin>>a[i][j];
k=1;
for(i=2;i<=n&& k;i++)
  for(j=1;j<i&&k;j++)
    if(a[i][j]) k=0;
if(k)
  cout<<"Este triunghiulara
superior";
else
  cout<<"Nu este triunghiulara
superior";
}
```

neintensiv

c

neintensiv

C/C++

for(i=0;i<n;i++) t[i]=s[i]; t[n]='\0';

Pascal

t:=copy(s,1,n);

Varianta PASCAL

```
Program var_69_II;
var k,n,i,j:integer;
    a:array[1..50,1..50] of integer;
begin
readln(n);
for i:=1 to n do
  for j:=1 to n do
    readln(a[i,j]);
k:=1;
for i:=2 to n do
  for j:=1 to i-1 do
    if a[i,j]<>0 then k:=0;
if k=1 then
  writeln('Este triunghiulara
superior')
else
  writeln('Nu este triunghiulara
superior')
end.
```

SUBIECTUL III

1. intensiv

a

2. intensiv

15 5 9 3 1

3. Varianta C/C++

```
int calcul(long n)
{int p=0,k=0;
While(n)
{
if (n%2==0) k++;
else p++;
n=n/10;
}
return abs(k-p);
}
```

4. a)

Varianta C/C++

```
#include<iostream.h>
#include<fstream.h>
ifstream f("numere.txt");
long n;
void afiseaza(int i)
{ long x;
if (i<=n)
{
f>>x;
if(x%2==0) cout<<x<<' \';
afiseaza(i+1);
}
if(x%2) cout<<x<<' \';
}
void main()
{
f>>n;
afiseaza(1);
}
```

neintensiv

a

neintensiv

Lipsește secvența 333

Varianta PASCAL

```
Function calcul(n:longint):integer;
var p,k:integer;
begin
p:=0; k:=0;
while (n<>0)do
begin
if n mod 2=0 then inc(k)
else inc(p);
n:=n div10;
end;
calcul:=abs(k-p);
end;
```

Varianta PASCAL

```
Program var_69_III;
var n:longint;
f:text;
procedure afiseaza(i:integer);
var x:longint;
begin
if i<=n then
begin
read(f,x);
if x mod 2=0 then
write(x,' ');
afiseaza(i+1);
end;
if x mod 2<>0 write(x,' ');
end;
begin
assign(f,'numere.txt'); reset(f);
readln(f,n);
afiseaza(1);
end.
```

b) Pentru a utiliza eficient memoria și timpul de executare am folosit un subprogram recursiv pentru citirea numerelor din fișier și, având în vedere faptul că ele sunt preluate în ordine crescătoare, elementele pare sunt afișate imediat după citire și înainte de autoapelul subprogramului, iar cele impare sunt afișate după încheierea autoapelurilor, fiind extrase din stivă în ordine inversă citirii lor, deci în ordine descrescătoare.

SUBIECTUL I

1. a

2.

b) $x=21, y=10$

c) **Varianta C/C++**

```
#include<iostream.h>
unsigned x,y,n;
void main(){
cin>>x>>y;
n=0;
while(x>=y){x=x-y; n++;}
cout<<n<<' \<<x;
}
```

Varianta PASCAL

```
Program var_70_I;
var x,y,n:word;
begin
readln(x,y);
n:=0;
while x>=y do begin x:=x-y; inc(n)
end;
writeln(n,' \',x);
end.
```

d) citește x,y

Repetă

```
| x ← x-y
| n ← n+1
|pana cand x<y
Scrie n,x
```

SUBIECTUL II

1.

a

2. intensiv

d

3. t: (2, 0, 2, 5, 2)

4. intensiv

C/C++

```
u1->next=p1;
```

Pascal

```
u1^.next:=p1;
```

neintensiv

d

neintensiv

C/C++

```
cout<<s[strlen(s)-1];
```

Pascal

```
writeln(s[length(s)]);
```

Varianta PASCAL

```
Program var_70_II;
var k,n,i,j,m:integer;
a:array[1..100,1..100] of
integer;
begin
readln(m,n,x);
for i:=1 to m do
for j:=1 to n do
readln(a[i,j]);
for i:=x to m-1 do
for j:=1 to n do
a[i,j]:=a[i+1,j];
for i:=1 to m do
begin
for j:=1 to n do
```

5. **Varianta C/C++**

```
#include<iostream.h>
int m,n,i,j,a[101][101],x;
void main()
{
cin>>m>>n>>x;
for(i=1;i<=m;i++)
for(j=1;j<=n;j++)
cin>>a[i][j];
for(i=x;i<=m;i++)
for(j=1;j<=n;j++)
a[i][j]=a[i+1][j];
m--;
for(i=1;i<=m;i++)
for(j=1;j<=n;j++)
cout<<a[i][j]<<' \';
cout<<endl;
```

```
}
}
```

```
write(a[i,j], ' ');
writeln
end;
end.
```

Explicații: Se elimină linia x, înlocuind toate liniile începând cu x până la penultima, cu liniile imediat următoare.

SUBIECTUL III

1. intensiv

d

2. intensiv

11

3. Varianta C/C++

```
void calcul(long n, long &x)
{x=0;
While(n)
{
if (n%2==0) x=x*10+n%10;
n=n/10;
}
}
```

4. a)

Varianta C/C++

```
#include<iostream.h>
#include<fstream.h>
ifstream f("numere.txt");
int n,i,j,max;
long x,y,k;
void main()
{
f>>n; max=0;
f>>x; i=1;
while (i<=n)
{
j=i+1; f>>y;
while(j<=n&&y==x){j++; f>>y;}
if(j-i>max){max=j-i;k=x;}
i=j; x=y;
}
cout<<k<<' '<<max;}
}
```

neintensiv

d

neintensiv

Lipsește secvența 444

Varianta PASCAL

```
procedure calcul(n:longint;var
x:longint);
begin
x:=0;
while n<>0 do
begin
if n mod 2=0 then x:=x*10+n mod
10;
n:=n div10;
end;
end;
```

Varianta PASCAL

```
Program var_70_III;
var x,y,k:longint;
f:text;
n,i,j,max:integer;
begin
assign(f,'numere.txt'); reset(f);
readln(f,n);
max:=0;
read(f,x); i:=1;
while i<=n do
begin
j:=i+1; read(f,y);
while (j<=n)and(y=x) do
begin
inc(j);read(f,y);
end;
if j-i>max then
begin
max:=j-i;k:=x
end;
i:=j; x:=y;
end;
writeln(k,' ',max);
end.
```

b) Se folosește faptul că șirul este ordonat crescător. Nu stocăm numerele citite într-un vector, ci vom folosi faptul că numerele sunt în ordine crescătoare, deci cele egale sunt dispuse pe poziții consecutive în fișier.

SUBIECTUL I

1. c.

2.

b) ~~420~~ 43, 26, 14

c)

Varianta C/C++

```
#include<iostream.h>
int n,i,j,x,s;
void main()
{
  cout<<"Dati n="; cin>>n;
  for(i=1; i<=n; i++)
  {
    cout<<"Dati x="; cin>>x;

    while(x>9)
      x=x/10;
    for(j=i; j<=i-1; j++)
      x=x*10;
    s=s+x;
  }
  cout<<s;
}
```

c) pseudocod:

s ← 0

citește n

pentru i ← 1, n execută

repetă x

dacă [x / 10] <> 0 atunci x ← [x / 10] altfel x ← x

până când x ≤ 9

pentru j ← 1, i-1 execută x ← x*10

s ← s+x

scrie s

Varianta PASCAL

```
var n,i,j,x,s:longint;
begin
  s:=0;
  write('Dati n='); readln(n);
  for i:=1 to n do
  begin
    write('Dati x='); readln(x);
    while x>9 do x:=x div 10;

    for j:=1 to i-1 do x:=x*10;
    s:=s+x;
  end;
  writeln(s);
  readln
end.
```

SUBIECTUL II

1. a. pentru intensiv respectiv c. pentru neintensiv

2. a. pentru intensiv respectiv d. pentru neintensiv

3. ~~s2:=s2+a[i][i];~~

4.

Varianta C/C++

```
gets(e1.ume); cin>>e1.nota;
gets(e2.ume); cin>>e2.nota;
if(e1.nota>=e2.nota)
  puts(e1.ume);
else
  puts(e2.ume);
```

5. Explicații

Folosind funcția POS, determinăm poziția subșirului c în șirul s, după care se face înlocuirea cu '*'

Varianta C/C++

```
#include<iostream.h>
```

s2:=s2+a[i,n-i+1];

Varianta PASCAL

```
readln(e1.ume); readln(e1.nota);
readln(e2.ume); readln(e2.nota);
If e1.nota>=e2.nota then
  writeln(e1.ume)
else
  writeln(e2.ume);
```

Varianta PASCAL

```
var s:string; c:string[20];
```

```
#include<string.h>
#include<stdio.h>
char s[20],c[20];
int m,n,i,p;
void main()
{
    cout<<"dati sirul s="; gets(s);
    cout<<"dati sirul c="; gets(c);
    n=strlen(s);
    m=strlen(c);
    p=*strstr(s,c);
    cout<<p<<endl;
    if(p==0)
        cout<<"NU APARE";
    else
    {
        while(p!=0)
        {
            for(i=p; i<=p+m-1; i++)

                p=*strstr(s,c);
            cout<<p<<endl;
        }
        puts(s);
    }
}
```

```

    m,n,i,p:integer;
begin
write('dati sirul s='); readln(s);
write('dati sirul c='); readln(c);
n:=length(s); m:=length(c);
p:=POS(c,s);
if p=0 then writeln('NU APARE');
    else
        begin
            while p<>0 do
                begin
                    for i:=p to p+m-1 do
                        s[i]:='*';
                    p:=pos(c,s);
                end;
                writeln(s);
            end;
        readln
    end.
end.
```

5. Explicații pentru neintensiv

Pentru a construi matricea cerută aceasta se parcurge pe coloane de sus în jos, iar la fiecare pas se actualizează valoarea care trebuie adăugată:

```

pentru j ← 1, n execută
    pentru i ← 1, n execută
        a[j,i] ← k;
        k ← k+1;
    sfârșit pentru
sfârșit pentru
```

SUBIECTUL III

1. d.

2. 5

3. Explicații

Funcția definită transformă numărul primit ca și parametru într-un șir de caractere, apoi se verifică dacă șirul astfel obținut este palindrom sau nu.

Varianta C/C++

```
int palindrom(long n)
{
    int nr,i;
    char s;
    nr=1;
    itoa(s,n,10);
    for(i=1; i<=(strlen(s)/2); i++)
        if(s[i]!=s[strlen(s)-i+1])
            nr=0;
    return nr;
}
```

Varianta PASCAL

```
function palindrom(n:longint):integer;
var nr,i:integer; s:string;
begin
    nr:=1;
    str(n,s);
    for i:=1 to length(s) div 2 do
        if s[i]<>s[length(s)-i+1] then nr:=0;
    palindrom:=nr;
end;
```

4. a)

Varianta C/C++

```
#include<iostream.h>
#include<fstream.h>

#include<string.h>
#include<values.h>
#include<stdio.h>
int n,nr,max;
int palindrom(long n)
{
    int nr,i;
    char *s;
    nr=1;
    itoa(n,s,10);
    for(i=1; i<=(strlen(s)/2); i++)
        if(s[i]!=s[strlen(s)-i+1])
            nr=0;
    return nr;
}
void main()
{
    ifstream f("numere.txt");
    max=-MAXINT;
    while(!f.eof())
    {
        f>>n;
        if(n>max && palindrom(n))
            max=n;
    }
    nr=0;
    f.close();
    ifstream h("numere.txt");
    while(!h.eof())
    {
        f>>n;
        if(n==max)
            nr++;
    }
    ofstream g("nrout.txt");
    g<<max; g<<nr;
    h.close(); g.close();
}
```

Varianta PASCAL

```
var n,nr,max:longint;f,g:text;
function palindrom(n:longint):integer;
var nr,i:integer; s:string;
begin
    nr:=1;
    str(n,s);
    for i:=1 to length(s) div 2 do
        if s[i]<>s[length(s)-i+1] then nr:=0;
    palindrom:=nr;
end;
begin
    assign(f,'c:\tp\numere.txt');reset(f);
    max:=-maxint;
    while not eoln(f) do
        begin
            read(f,n);
            if (n>max) and (palindrom(n)=1) then
                max:=n;
        end;
        reset(f);
        nr:=0;
        while not eoln(f) do
            begin
                read(f,n);
                if n=max then nr:=nr+1;
            end;
        assign(g,'c:\tp\nrout.txt');rewrite(g);
        writeln(g,max);writeln(g,nr);
        close(f);close(g);
    end.
```

b) Programul citește din fișierul text *numere.in* numerele, iar funcția *palindrom* se apelează doar pentru numerele mai mari decât *max*.

Varianta <72>

SUBIECTUL I

1. d.

2. a)

**

*

**

b) ($j \geq 0$) and ($j < 4$)

c)

Varianta C/C++

```
#include<iostream.h>
int n,i,j,b;
void main()
{
    cout<<"dati n=" ; cin>>b ;
    for(i=1 ; i<=2*n-1 ; i++)
    {
        b=0 ;
        if(n-i<0)
            j=i-n ;
        else
            j=n-i ;
        while(j>=0)
        {
            j=j-1;
            b=1;
        }
        if(b==0)
            cout<<"\n";
    }
}
```

Varianta PASCAL

```
var n,i,j,b:integer;
begin
    write('dati n='); readln(n);
    for i:=1 to 2*n-1 do
    begin
        b:=0;
        if (n-i)<0 then j:=i-n else j:=n-
        i;
        while j>=0 do
        begin
            write('*');
            j:=j-1;
            b:=1;
        end;
        if b=1 then writeln;
    end;
end.
```

SUBIECTUL II

1. c. **pentru intensiv**, respectiv c. **pentru neintensiv**

2. d. **pentru intensiv**, respectiv a. **pentru neintensiv**

3. graful dat are o componentă conexă, prin eliminarea vârfului 1 se obține un subgraf cu 3 componente conexe (maxim în acest caz).

4. **Pentru intensiv:**

$p^{\wedge}.info:=3;$

$u^{\wedge}.info:=1;$

Pentru neintensiv:

Prin adăugarea muchiilor (3,1) respectiv (4,5) între oricare două noduri x și y există cel puțin un drum. (număr minim de muchii).

5. **Explicații pentru intensiv**

Matricea se construiește pornind de la primul element egal cu o valoare *val* inițial egală cu 1, care apoi la fiecare pas crește cu o unitate. Se parcurge matricea de sus în jos pe semidiagonale până la diagonală secundară, apoi restul elementelor aflate sub diagonală secundară vor avea rînd valori egale cu elementele deja construite astfel asigurându-se simetria.

Varianta C/C++

```
#include<iostream.h>
int a[100][100],i,n,j,x,y,val;
void main()
{
    cout<< »dati n= » ; cin>>n ;
    val=1;
    for(x=1; x<=n; i++)
    {
```

Varianta PASCAL

```
var a:array[1..100,1..100] of
integer;
    i,n,j,x,y,val:integer;
begin
    write('dati n='); readln(n);
    val:=1;
    for x:=1 to n do
    begin
```

```

i=x ; j=1 ;
for(y=1; y<=x; y++)
{
    a[i][j]=val; i--; j++; val++;
}
}
y=1;
for(i=n; i>=2; i--)
{
    x=1;
    for(j=n; j>=y+1; j--)
        {a[i][j]=a[x][y]; x++;}
    y++;
}
for(i=1; i<=n; i++)
{
    for(j=1; j<=n; j++)
        cout<<a[i][j]<<" ";

}cout<<"\n";
}

```

```

i:=x; j:=1;
for y:=1 to x do
begin
    a[i,j]:=val; i:=i-1; j:=j+1;
val:=val+1;
end;
end;
y:=1;
for i:=n downto 2 do
begin
    x:=1;
    for j:=n downto y+1 do
        begin a[i,j]:=a[x,y];
x:=x+1;end;
    y:=y+1;
end;
for i:=1 to n do
begin
    for j:=1 to n do
        write(a[i,j], ' ');
    writeln;
end;
end.

```

5. Explicații pentru neintensiv:

Se construiește matricea parcurgând în ordine toate elementele de deasupra diagonalei secundare inclisiv ($n+1 \leq i+j$), atribuind o valoare $k=1,2,3,\dots$, apoi respectând condiția de simetrie (vezi varianta intensiv) se atribuie elementelor de sub diagonala secundară valorile corespunzătoare.

ȘUBIECTUL III

1.0.

2. 111001

3. varianta intensiv:

Varianta C/C++

```

void ecuatie(int a, int b, int c)
{
    float d, x1, x2;
    if (a!=0)
    {
        d=b*b-4*a*c;
        if (d>0)
        {
            x1=(-b+sqrt(d))/(2*a);
            x2=(-b-sqrt(d))/(2*a);
            if (x1>0)
                cout<<x1:5:2;
            else
                cout<<x2:5:2;
            else
                cout<<"-32000";
        }
    }
    if (d==0)
    {
        x1=-b/(2*a);
        if (x1>0)
            cout<<x1:5:2;
        else

```

Varianta PASCAL

```

procedure ecuatie(a,b,c:integer);
var d,x1,x2:real;
begin
    if a<>0 then
        begin
            d:=b*b-4*a*c;
            if d>0 then
                begin
                    x1:=(-b+sqrt(d))/(2*a);
                    x2:=(-b-sqrt(d))/(2*a);
                    if (x1>0) or (x2>0) then
                        if x1>x2 then writeln(x1:5:2)
                    else writeln(x2:5:2)
                    else write(-32000);
                end;
            if d=0 then
                begin
                    x1:=-b/(2*a);
                    if x1>0 then writeln(x1:5:2)
                end;
            if d<0 then writeln(-32000);
        end;
end.

```

```

        cout<<"-32000";
    }
    if(d<0)
        cout<<"-32000";
    }
    else
        cout<<"a=0";
}

```

```

else writeln('a=0');
end;

```

4.

b) Se observă că $k = n(n+1)/2 + x$, deoarece până la poziția k sunt $1+2+\dots+n$ termeni plus un x care indică numărul corespunzător din subșirul $1,2,3,4,\dots$.

Astfel, calculăm rădăcina pozitivă a ecuației $n^2 + n - 2k = 0$ (considerăm $x=0$).

Dacă aceasta este un număr întreg, atunci acesta este numărul căutat. În caz contrar, numărul căutat se află în subșirul $1,2,3,\dots,n+1$. Pentru a afla care este acesta calculăm $(x - [x]) \cdot (x+1)$ (partea fracționară a lui x împarte intervalul $[x, x+1]$ în $(x+1)$ părți egale).

3. varianta neintensiv:

Varianta C/C++

```

int cifre(long a, long b)
{
    long x,y;
    int i,cx,cy,c;
    c=0;
    for(i=1; i<=9; i++)
    {
        x=a; y=b; cx=0; cy=0;
        while(x!=0)

            if(x%10!=i)
            else
            {
                cx=1; x=0;
            }
        while(y!=0)
            if(y%10!=i)
                y=y/10;
            else
            {
                cy=1; y=0;
            }
        if(cx==1 && cy==1)
            c++;
    }
    cifre=c;
}

```

Varianta PASCAL

```

function cifre(a,b:longint):integer;
var x,y:longint; i,cx,cy,c:integer;
begin
    c:=0;
    for i:=1 to 9 do
    begin
        x:=a; y:=b;cx:=0;cy:=0;
        while (x<>0) do
            if (x mod 10<>i) then x:=x div 10

        while (y<>0) do
            if (y mod 10<>i) then y:=y div 10
            else begin cy:=1; y:=0; end;
            if (cx=1) and (cy=1) then c:=c+1;
        end;
    cifre:=c;
end;

```


SUBIECTUL I

1. c.

2.
a) 12

b) 125

c)

Varianta C/C++

```
#include<iostream.h>
int x,i,a,b,p,nr;
void main()
{
    cin>>a>>b>>c;
    if(a>b)
        {x=a; a=b; b=x;}
    else
        {
            nr=0;
            for(i=a; i<=b; i++)
                {
                    x=i;
                    while(x!=0 && x%p!=0)
                        x=x/10;
                    if(x!=0)
                        nr++;
                }
            cout<<nr;
        }
}
```

d)

```
citește a,b,p
nr←0
pentru i←a,b execută
    x←i
    repetă
        x←[x/10]
        până când (x=0) or (x%p=0)
        dacă x≠0 atunci
            nr←nr+1
scrie nr
```

Varianta PASCAL

```
var x,i,a,b,p,nr:integer;
begin
    readln(a, b, p);
    if a>b then
        begin x:=a; a:=b; b:=x; end
    else
        begin
            nr:=0;
            for i:=a to b do
                begin
                    x:=i;
                    while (x<>0) and (x mod p<>0) do
                        x:=x div 10;
                    if x<>0 then nr:=nr+1;
                end;
            writeln(nr);
        end;
end.
```

SUBIECTUL II

1. b.

2. c.

3. Numărul minim de arce este 2; exemplu de circuit elementar de lungime 4: (1,2)(2,3)(3,4)(4,1); unde (3,4)(4,1) sunt cele două arce adăugate.

4.

Varianta C/C++

```
#include<iostream.h>
int n,p,q,l,aux;
.....
for(i=1; i<=n; i++)
```

Varianta PASCAL

```
Var n,p,q,l,aux:integer;
.....
For i:=1 to n do
    Begin
```

```
{
  aux=a[p][i];
  a[p][i]=a[q][i];
  a[q][i]=aux;
}
```

```
aux:=a[p,i];
a[p,i]:=a[q,i];
a[q,i]:=aux;
End;
```

5. Explicații

Explicații necesare înțelegerii codului

Varianta C/C++

```
#include<iostream.h>
#include<string.h>
#include<stdio.h>
int n,i,nrvoc,min,j,p;
char s[101];
void main()
{
  cout<<"dati nr de cuvinte=";
  cin>>n;
  for(i=1; i<=n; i++)
    cin>>s[i];
  p=0; min=255;
  for(i=1; i<=n; i++)
  {
    nrvoc=0;
    for(j=1; j<=strlen(s[i]); j++)
      if(s[i][j]== "a" || s[i][j]==
"e" || s[i][j]== "i" || s[i][j]==
"o" || s[i][j]== "u")
        nr++;
    if(nrvoc<min)
      min=nrvoc;

    if(nrvoc==min)
      p++;
    if(p!=0)
      puts(s[p]);
  }
}
```

Varianta PASCAL

```
var n,i,nrvoc,min,j,p:integer;
s:array[1..101] of string;
begin
write('dati nr de cuvinte=');
readln(n);
for i:=1 to n do
  readln(s[i]);
p:=0;min:=255;
for i:=1 to n do
  begin
    nrvoc:=0;
    for j:=1 to length(s[i]) do
      if s[i][j] in
['a','e','i','o','u']
then nrvoc:=nrvoc+1;
if nrvoc<min then min:=nrvoc;
if nrvoc==min then p:=i;
end;
if p<>0 then writeln(s[p]);
end.
```

5. varianta neintensiv

Se parcurge șirul citit pornind de la primul până la ultimul caracter, la fiecare pas se testează dacă $s[i]=' '$ (adică, caracterul i din șirul s este egal cu caracterul spațiu), dacă da, atunci $nr \leftarrow nr+1$ (nr reprezintă numărul de cuvinte din șir, inițial 0) și $s[i+1]$ în $['a','e','i','o','u']$ atunci $nrv \leftarrow nrv+1$ (unde nrv este numărul de cuvinte care încep cu o

vocală, inițial egal cu 0). Dacă primul caracter din șir este și vocală sau 0 dacă primul caracter din șir nu este vocală). La sfârșitul șirului se afișează și nrv .

SUBIECTUL III

1. a.

2. 137486

3. a) Cautăm poziția de inserat prin metoda *divide et impera*, la fel pentru **varianta neintensiv**

Varianta C/C++

```
int cauta (int n, int x, int val)
{
  int i;
  i=1; s=n; ok=0;
  while(i<=s && ok==0)
  {
```

Varianta PASCAL

```
Function cauta
(n:integer;x:vector;val:integer):integ
er;
var i:integer;
begin
```

```

m=(i+s)/2;
if(val>x[m] && val<x[m+1])
    ok=1;
if(val>x[m])
    i=m++;
else
    i=m--;
}

```

```

i:=1;s:=n;ok:=false;
while (i<=s) and (ok=false) do
begin
    m:=(i+s) div 2;
    if (val>x[m]) and (val<x[m+1]) then
ok:=true;
    if (val>x[m]) then i:=m+1 else s:=m-1;
end;
if ok then cauta:=m+1;
end ;

```

Varianta <74>

SUBIECTUL I

1. a.
- 2.

b) $q=2345, p=322$

c)

Varianta C/C++

```

#include<iostream.h>
int a,b,c,p;
void main()
{
    cin>>a>>b;
    c=0; p=0;
    while ((a+b)>10)
    {
        if ((a%10==b%10) && (a%10%2==1))
            c=c*10+b%10;
        else
            p=p*10+a%10;
        a=a/10;
        b=b/10;
    }
    cout<<c<<" "<<p
}

```

Varianta PASCAL

```

var a,b,c,p:longint;
begin
    readln(a,b);
    c:=0; p:=0;
    while (a+b)>10 do
        begin
            if (a mod 10=b mod 10) and (a mod 10 mod 2=1) then
                c:=c*10+b mod 10; else
                p:=p*10+a mod 10;
            a:=a div 10;
            b:=b div 10;
        end;
    writeln(c, ' ', p);
    readln
end.

```

SUBIECTUL II

1. a. **pentru intensiv** respectiv a. **pentru neintensiv**
2. c. **pentru intensiv** respectiv c. **pentru neintensiv**

3. a) Descendenții (fii) nodului 2 sunt: 4,6,9

b) Arborele are 5 noduri de tip frunză.

4. a) Numărul minim de muchii care trebuie eliminate a.î. graful parțial obținut sa nu aibă nici un ciclu este 2.

b) O singurămuchie (2,3), trebuie eliminat aa.î. graful parțial obținut să aibă exact două componente conexe.

5. Explicații

Funcția *FIB* primește ca și parametru un număr reprezentând poziția termenului în șirul lui

Fibonacci și returnează valoarea termenului de pe poziția respectivă. Apelând această funcție se construiește matricea cerută.

Varianta C/C++

```
#include<iostream.h>
int n,i,j,k,a[50][50];
long fib(int k)
{
    if(k==1 || k==2)
        fib=1;
    else
        fib=fib(k-1)+fib(k-2);
}
void main()
{
    cout<<"dati n="; cin>>n;
    k=1;
    for(i=1; i<=n; i++)
        for(j=1; j<=n; j++)
            {
                a[i][j]=fib(k)%10;
                k++;
            }
    for(i=1; i<=n; i++)
        {
            for(j=1; j<=n; j++)
                cout<<a[i][j]<<" ";
            cout<<"\n";
        }
}
```

Varianta PASCAL

```
var n,i,j,k:integer;
a:array[1..50,1..50] of integer;
function fib(k:integer):longint;
begin
    if (k=1) or (k=2) then fib:=1
        else fib:=fib(k-1)+fib(k-2);
end;
begin
    write('Dati n='); readln(n);
    k:=1;
    for i:=1 to n do
        for j:=1 to n do
            begin
                a[i,j]:=fib(k) mod 10;
                k:=k+1;
            end;
        end;
    for i:=1 to n do
        begin
            for j:=1 to n do
                write (a[i,j], ' ');
            writeln;
        end;
    end.
```

5. Explicații varianta neintensiv

Se construiește matricea folosind algoritmul alăturat :

pentru $i \leftarrow 1, n$ execută

pentru $j \leftarrow 1, n$ execută

dacă $(i \bmod 2 = 0)$ atunci $a[i,j] \leftarrow n+1-j$ altfel $a[i,j] \leftarrow j$

Apoi se afișează matricea astfel construită.

SUBIECTUL III

1. b. pentru intensiv, respectiv c. pentru neintensiv

2. 32, -60

3.

Varianta C/C++

```
int cifra(int n)
{
    int n;
    x=n%10;
    while(n!=0)
        {
            if(n%10>x)
                x=n%10;
        }
    cifra=x;
}
```

Varianta PASCAL

```
function cifra(n:integer):integer;
var x:integer;
begin
    x:=n mod 10;
    while n<>0 do
        begin
            if n mod 10>x then x:=n mod 10;
                n:=n div 10;
        end;
    cifra:=x;
end;
```

4.

a)

Varianta C/C++

```
#include<iostream.h>
#include<fstream.h>
```

Varianta PASCAL

```
type vector=array[1..20] of integer;
var v:vector;
```

```

int v[20],nr,n,i,x,aux,ok;
int cifra(int n)
{
    x=n%10;
    while(n!=0)
    {
        if(n%10>x)
            x=n%10;
        n=n/10;
    }
    cifra=x;
}
void main()
{
    ifstream f("c:\numere.in");
    ofstream g("c:\numere.out");
    f>>n;
    for(i=1; i<=n; i++)

        {f>>x;
        v[i]=cifra(x);
        }
    do
    {
        ok=1;
        for(i=1; i<=n-1; i++)
            if(v[i]>v[i+1])
            {
                aux=v[i];
                v[i]=v[i+1];
                v[i+1]=aux;
            }
        ok=0;
    }
    while(ok)
    i=1;
    while (v[i]==0)
        i++;
    aux=v[i]; v[1]=v[i]; v[i]=aux;
    for(i=1; i<=n; i++)
        nr=nr*10+v[i];
    g<<nr;
    f.close();
    g.close();
}

```

```

nr,n,i,x,aux:integer;
ok:boolean; f,g:text;
function cifra(n:integer):integer;
var x:integer;
begin
    x:=n mod 10;
    while n<>0 do
    begin
        if n mod 10>x then x:=n mod 10;
        n:=n div 10;
    end;
    cifra:=x;
end;
begin
    assign(f,'c:\numere.in');
    reset(f);
    assign(g,'c:\numere.out');
    rewrite(g);

    readln(f,n);
    for i:=1 to n do
    begin
        read(f,x);
        v[i]:=cifra(x);
    end;
    repeat
        ok:=true;
        for i:=1 to n-1 do
            if v[i]>v[i+1] then
            begin
                aux:=v[i];
                v[i]:=v[i+1];
                v[i+1]:=aux;
                ok:=false;
            end;
        until ok;
        i:=1;
        while v[i]=0 do i:=i+1;
        aux:=v[1]; v[1]:=v[i]; v[i]:=aux;
        for i:=1 to n do nr:=nr*10+v[i];
        writeln(g,nr);
        close(f);
        close(g);
    end.

```

4. penru neintensiv

a) Se aplică funcția *cifra* pentru numerele citite pe rând din fișier, iar pentru fiecare număr citit se apelează funcția *cifra* pentru a determina dacă cifra maximă este egală cu cifra *c*, o variabilă contor va memora câte dintre numerele citite îndeplinesc această condiție, rezultatele cerute se vor scrie în fișier.

Varianta <75>

SUBIECTUL I

1. d.

2.

a) 62255661

b) 123, 3456, 5698, 7832

c)

Varianta C/C++

```
#include<iostream.h>
int a, k;

long x;
void main()
{
    a=0; k=0;
    do
    {
        cout<<"dati x="; cin>>x;
        while(x>99)
            x=x/10;
        if(x>9)
        {
            a=a*100+x;
            k++;
        }
    }
    while(k>4)
    cout<<a;
}
```

Varianta PASCAL

```
var x:longint;a,k:integer;
begin
    a:=0; k:=0;
    repeat
        write('Dati x=');readln(x);
        while x>99 do x:=x div 10;
        if x>9 then
            begin
                a:=a*100+x;
                k:=k+1;
            end;
    until k=4;
    writeln(a);
    readln
end.
```

d) pseudocod

```
a ← 0
k ← 0
cât timp k < 4 execută
    citește x (număr natural)
    cât timp x > 99 execută x ← [x/10]
    dacă x > 9 atunci
        a ← a * 100 + x
        k ← k + 1
    sfarsit_daca
sfarsit_cat_timp
sfarsit_cat_timp
scrie a
```

SUBIECTUL II

1. b. pentru intensiv respectiv d. pentru neintensiv

3. pentru intensiv
r[^].urm:=q

3. pentru neintensiv
ADD5, ELIM

4. Numărul minim de arce care trebuie adăugate este 2, arcele sunt: (2,3) și (4,6)

5.

Varianta C/C++

```
#include<iostream.h>
int a[100][100],i,n,j,k,val;
void main()
{
    cout<<"dati n="; cin>>n;
    for(i=1; i<=n; i++)
        for(j=1; j<=n; j++)
            a[i][j]=0;
}
```

Varianta PASCAL

```
var a:array[1..100,1..100] of
integer;
    i,n,j,k,val:integer;
begin
    write('dati n=');readln(n);
    for i:=1 to n do
        for j:=1 to n do a[i,j]:=0;
    val:=2;
```

<pre> val=2; for(k=1; k<=2*n-1; k++) { for(i=1; i<=n; i++) for(j=1; j<=n; j++) if(i+j==k+1) a[i][j]=val*val; val++; } for(i=1; i<=n; i++) { for(j=1; j<=n; j++) cout<<a[i][j]<< " "; cout<<"\n"; } </pre>	<pre> for k:=1 to 2*n-1 do begin for i:=1 to n do for j:=1 to n do if i+j=k+1 then a[i,j]:=val*val; val:=val+1; end; for i:=1 to n do begin for j:=1 to n do write (a[i,j], ' '); writeln; end; end. </pre>
--	---

5. pentru neintensiv

Se inițializează matricea cu valoarea zero, apoi se aplică următorul algoritm :

```

i ← 3
cât timp i ≤ m execută
  pentru j ← 1, n execută a[i,j] := 1;
  pentru j ← 1, n execută a[i+1,j] := 1;
  i := i + 2;
sfârșit cât timp.

```

Apoi se afișează matricea astfel construită.

SUBIECTUL III

1. $d_6, -2, 0, 5, 10, 7$ pentru intensiv, respectiv $-2, 0, 7, -6, 5, 10$ pentru neintensiv

3. a)

Varianta C/C++

```

long ultimacifra (int a, int b)
{
  int u,v;
  p=1;
  u=a;
  v=b;
  while (v>0)
    if (v%2==0)
      {
        u=u*u; v=v/2;
      }
    else
      {
        p=p*u; v=v-1;
      }
  ultimacifra=p%10;
}

```

Varianta PASCAL

```

function
ultimacifra(a,b:integer):longint;
var u,v:integer;
begin
  p:=1;
  u:=a;
  v:=b;
  while v>0 do
    if v mod 2 =0 then
      begin
        u:=u*u; v:=v div 2;
      end
    else
      begin
        p:=p*u; v:=v-1;
      end;
  ultimacifra:=p mod 10;
end;

```

b) Pentru rezolvare pornim de la condiția

$$I := p * u^v = a^b$$

pe care o vom menține adevărată pe timpul execuției subprogramului. Inițial ea devine adevărată dacă $\hat{a}p=1, u=a$ și $v=b$. Execuția trebuie să se termine când $v=0$ deci I devine $p = a^b$.

4.

Varianta C/C++

```
#include<iostream.h>
#include<fstream.h>
int i,n,s;

long ultimacifra (int a, int b)
{
    int u,v;
    p=1;
    u=a;
    v=b;
    while (v>0)
        if (v%2==0)
            {
                u=u*u; v=v/2;
            }
        else
            {
                P=p*u; v=v-1;
            }
    ultimacifra=p%10;
}

void main()
{
    ifstream f("c:\bc\sirin.txt");
    ofstream g("c:\bc\sirout.txt");
    f>>n; s=0;
    for(i=1; i<=n; i++)
        {
            f>>a>>b;
            s=s+ultimacifra(a,b);
        }
    cout<<q<<s;
    f.close();
    g.close();
}
```

Varianta PASCAL

```
var p,a,b:longint;
f,g:text;i,n,s:integer;
function
ultimacifra(a,b:integer):longint;
var u,v:integer;
begin
    p:=1;
    u:=a;
    v:=b;
    while v>0 do
        if v mod 2 =0 then
            begin
                u:=u*u; v:=v div 2;
            end
        else
            begin
                p:=p*u;
                v:=v-1;
            end;
    ultimacifra:=p mod 10;
end;
begin
    assign(f,'c:\tp\sirin.txt');reset(f);
    assign(g,'c:\tp\sirout.txt');rewrite(g);
    readln(f,n);s:=0;
    for i:=1 to n do
        begin
            readln(f,a,b);
            s:=s+ultimacifra(a,b);
        end;
    writeln(g,s);
    close(f); close(g);
    readln
end.
```

Varianta <76>

SUBIECTUL I

1. c.

2.

a) 35
b) De exemplu un număr prim: 2

c)

Varianta C/C++

```
#include<iostream.h>
int a,x,p,c;
void main()
{
    cout<<"dati a="; cin>>a;
    x=2; p=1;
    while (a>1)
        {
            c=0;
            while (a%x==0)
                {
                    c=x;
                }
        }
}
```

Varianta PASCAL

```
var a,x,p,c:integer;
begin
    write('dati a=');readln(a);
    x:=2;p:=1;
    while a>1 do
        begin
            c:=0;
            while a mod x=0 do
                begin
                    c:=x;
                    a:=a div x;
                end;
        end;
```



```

    a=a/x;
    }
    if(c!=0)
        p=p*c;
        x++;
    }
    cout<<p;
}

```

```

    if c<>0 then p:=p*c;
    x:=x+1;
end;
writeln(p);
end.

```

d)

citește a (număr natural)
y←2
p←1
repetă
 c←0
 dacă a/n atunci
 repetă
 c←x

 a← $\lfloor a/x \rfloor$
 până când a%x<>0
 dacă c<>0 atunci
 p←p*c
 x←x+1
până când a<=1
scrie p

SUBIECTUL II

1. b.

2. b.

3. Num rul minim de arce este 3, **pentru intensivă** | 3. Valoarea 2, **pentru neintensiv**

4. Vectorul de tați este: 5 3 6 6 7 0 6 3

5. Explicații

Se extrage din șirul inițial un subșir începând cu poziția 2 de lungime length(s)-2, unde s este șirul prelucrat anterior, la fiecare pas se afișează sub șirul astfel obținut.

Varianta C/C++

```

#include<iostream.h>
#include<string.h>
#include<stdio.h>
char s[20];
int i,n;
void main()
{
    gets(s);
    puts(s);
    i=strlen(s);
    while(s!=" ")
    {
        s=strncpy(s,2,strlen(s)-2)
        puts(s);
    }
}

```

Varianta PASCAL

```

var s:string[20]; n,i:integer;
begin
    readln(s);
    writeln(s);
    i:=length(s);
    While s<>' ' do
    begin
        s:=copy(s,2,length(s)-2);
        writeln(s);
    end;
end.

```

SUBIECTUL III

1. d.

2. 21

3.

Varianta C/C++

```
int s(long n; int c)
{
    int k;

    while(n!=0)
    {
        if(n%10>=c-1 && n%10<=c+1)
            k++;
        n=n/10;
    }
    s=k;
}
```

Varianta PASCAL

```
Function
s(n:longint;c:integer):integer;
Var k:integer;

begin
    while n<>0 do
        begin
            If (n mod 10 >=c-1) and (n mod 10
            <= c+1) then k:=k+1;
                n:=n div 10;
        end;
    s:=k;
end;
```

4.

a) O metodă eficientă de căutare într-un vector sortat crescător este *divide et impera*.

b) Procedura caută, caută elementul k, în vectorul v, cu n elemente.

Varianta C/C++

```
#include<iostream.h>
int n,m,i,k,ok,v[20],u[20];
void cauta(int k, int n, int v[20])
{
    int inc,sf,mij;
    inc=1; sf=n;
    ok=0;
    while(inc<=sf && ok==0)
    {
        mij=(inc+sf)/2;
        if(v[mij]==k)
            ok=1;
        if(k<v[mij])
            sf=mij-1;
        else
            inc=mij+1;
    }
    if(ok)
        cout<<k<<"se afla pe
        pozitia"<<mij;
    else
        cout<<"nu am gasit elementul
        cautat";
}
void main()
{
    ifstream f("bac.txt");
    f>>n>>m;
    for(i=1; i<=n; i++)
        f>>v[i];
    for(i=1; i<=m; i++)
        f>>u[i];
    for(i=1; i<=m; i++)
        cauta(u[i],n,v);
    f.close();
}
```

Varianta PASCAL

```
type vector=array[1..20]of integer;
var n,m,i,k:integer; ok:boolean;
f:text; v,u:vector;
procedure
cauta(k,n:integer;v:vector);
var inc,sf,mij:integer;
begin
    inc:=1;sf:=n;
    ok:=false;
    while (inc<=sf)and(ok=false)do
        begin
            mij:=(inc+sf) div 2;
            if v[mij]=k then ok:=true;
            if k<v[mij] then sf:=mij-1
                else inc:=mij+1;
        end;
    if ok then writeln(k,'se afla pe
    pozitia',mij)
        else writeln('nu am gasit
        elemntul cautat');
    end;
begin
    assign(f,'bac.txt');reset(f);
    readln(f,n,m);
    for i:=1 to n do read(f,v[i]);
    readln(f);
    for i:=1 to m do read(f,u[i]);
    for i:=1 to m do cauta(u[i],n,v);
    close(f);
end.
```

SUBIECTUL I

1. b.

2.

b) Un exemplu de date : 7 6 3 0

c)

Varianta C/C++

```
#include<iostream.h>
int a,b,k;
void main()
{
    cin>>a;
    k=0;
    while(a!=0)
    {
        cout<<"dati b="; cin>>b;
        if(a<b)
            k++;
        a=b;
    }
    cout<<k;
}
```

d)

```
cirește a
k←0
repetă
    cite te b
    dacă a<b atunci k←k+1
    a←b
până când a=0
scrie k
```

SUBIECTUL II

1. a.

2. a. pentru intensiv, respectiv c. pentru neintensiv

3. pentru intensiv : Un număr de 2 arce trebuie adăugate pentru ca toate vârfurile să

4. pentru intensiv : Pentru un graf conex cu 21 noduri și 20 muchii sunt 2 noduri de grad 1.

5.

Varianta C/C++

```
#include<iostream.h>
#include<string.h>
#include<stdio.h>
char s[50];
int n,i;
void main()
{
    gets(s); n=strlen(s);
    s[1]=s[1]+32;
    for(i=2; i<n; i++)
        if(s[i]==" ")
```

Varianta PASCAL

```
var a,b,k:integer;
begin
    readln(a);
    k:=0;
    while a<>0 do
        begin
            write('Dati b=');readln(b);
            if a<b then k:=k+1;

            a:=b;
        end;
    writeln(k);
end.
```

3. pentru neintensiv : Numărul de noduri de grad 1 este 3

4. pentru neintensiv : Valoarea din vârful stivei este 5.

Varianta PASCAL

```
Var s:string[50]; n,i:integer;
Begin
    Readln(s);n:=length(s);
    s[1]:=upcase(s[1]);
    For i:=2 to n-1 do
        If s[i]=' ' then
            begin
                s[i-1]:=upcase(s[i-1]);
                s[i+1]:=upcase(s[i+1]);
            end;
    s[n]:=upcases(s[n]);
```

```

{
    s[i-1]=s[i-1]+32;
    s[i+1]=s[i+1]+32;
}
s[n]=s[n]+32;
puts(s);
}

```

```

writeln(s);
end.

```

SUBIECTUL III

1. a.

2. 10

4. După citirea datelor din fișier se formează vectorul x care se ordonează crescător, după care se parcurge vectorul x comparându-se componenta curentă cu cea imediat următoare. Odată cu această parcurgere se construiește vectorul y de componente distincte și cel cu frecvențele de apariție. Se afișează componentele din vectorul y , pentru care frecvența este

maximă. Eficiența constă în parcurgerea vectorului de componente o singură dată pentru a afla elementele distincte împreună cu frecvența de apariție.

b)

Varianta C/C++

```

#include<iostream.h>
#include<fstream.h>
int x[100],y[100],f[100],n,k,i,aux;
int max,ok;
void main()
{
    ifstream g("bacin.txt");
    g>>n;

    for(i=1; i<=n; i++)
        g>>x[i];
    do
    {
        ok=1;
        for(i=1; i<n; i++)
            if(x[i]>x[i+1])
            {
                aux=x[i]; x[i]=x[i+1];
                x[i+1]=aux; ok=0;
            }
    }
    while(ok)
    k=1;
    y[k]=x[1];
    f[k]=1;
    for(i=1; i<n; i++)
        if(x[i]!=x[i+1])
        {
            k++;
            y[k]=x[i+1];
            f[k]=1;
        }
    else
        f[k]=f[k]++;

    max=f[1];
    for(i=1; i<=k; i++)
        if(f[i]>max)
            max=f[i];
}

```

Varianta PASCAL

```

var x,y,f:array[1..100] of integer;
g:text;
n,k,i,aux,max:integer; ok:boolean;
begin
    assign(g,'bacin.txt');reset(g);
    readln(g,n);
    for i:=1 to n do
        read(g,x[i]);

    repeat
        ok:=true;
    for i:=1 to n-1 do
        if x[i]>x[i+1] then
            begin
                aux:=x[i]; x[i]:=x[i+1];
                x[i+1]:=aux;
                ok:=false;
            end;
    until ok;
    k:=1;
    y[k]:=x[1];
    f[k]:=1;
    for i:=1 to n-1 do
        if x[i]<>x[i+1] then
            begin
                k:=k+1;
                y[k]:=x[i+1];
                f[k]:=1;
            end
            else f[k]:=f[k]+1;
    max:=f[1];
    for i:=1 to k do
        if f[i]>max then max:=f[i];
    for i:=1 to k do
        if f[i]=max then write(y[i],' ');
    close(g);
end.

```

```

for(i=1; i<=n; i++)
  if(f[i]=max)
    cout<<y[i]<<" ";
g.close();
}

```

Varianta <78>

SUBIECTUL I

1. a.

2.

a) 3

b) Orice șir de cel puțin 3 numere care nu are două numere consecutive cu ultimele cifre egale

Exemplu: 12 23 34 45 56 0

c)

Varianta C/C++

```

#include<iostream.h>
void main()
{
  int a,k=0,b;
  cin>>a;
  while(a)
  {
    cin>>b;
    if(a%10==b%10) k++;
    a=b;
  }
  cout<<k;
}

```

Varianta PASCAL

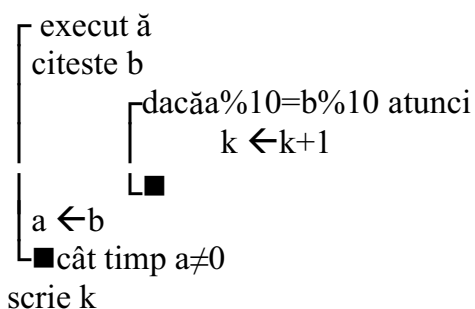
```

Var a,b,k:integer;
Begin
k:=0;
readln(a);
while (a<>0) do
  begin
  read(b);
  if a mod 10=b mod 10 then inc(k);
  a:=b;
  end;
writeln(k);
end.

```

d) citește a

$k \leftarrow 0$



SUBIECTUL II

1. b

2. **NOTA!** Varianta corectă nu se regăsește printre cele posibile. 1 2 4 5 6

Pentru neintensiv varianta a

3. 2

4. 99 pentru intensiv, respectiv 4 pentru neintensiv

5.

Varianta C/C++

Varianta PASCAL

```
#include<iostream.h>
#include<string.h>
void main()
{
    char s[100],cuv[15],*p,sep[]=" ";
    cin.get(s,100);
    cin.get();
    cin>>cuv;
    p=strtok(s,sep);
    while(p)
    {
        if(strstr(p,cuv)) cout<<p<<"? ";
        else cout<<p<<" ";
        p=strtok(NULL,sep);
    }
}
```

```
var sir,s1,csir:string[100];
    s:string[15];
    k,i:integer;
begin
    readln(sir);
    csir:=sir;
    readln(s);
    i:=0;
    while (pos(' ',sir)<>0) do
        begin
            s1:='';
            k:=pos(' ',sir);
            s1:=copy(sir,1,k-1);
            i:=i+k;
            if pos(s,s1)<>0 then
                begin
                    insert('?',csir,i);
                    inc(i);
                end;
            delete(sir,1,k);
        end;
    if pos(s,sir)<>0 then
        insert('?',csir,i+length(sir)+1);
    writeln(csir);
end.
```

SUBIECTUL III

1. c

2. 3

3. a)pentru intensiv

Varianta C/C++

```
void prim(int n, int &p)
{
    int i;
    p=1;
    if (n==0||n==1) p=0;
    else
        for(i=2;i<=n/2&&p;i++)
            if (n%i==0) p=0;
}
```

b)pentru intensiv

Varianta C/C++

```
#include<iostream.h>
void prim(int n, int &p)
{
    int i;
    p=1;
    if (n==0||n==1) p=0;
    else
        for(i=2;i<=n/2&&p;i++)
            if (n%i==0) p=0;
}
void main()
{
    int n;
    while(n<n)
    {
        int i=0, j=2, ok;
        while(i<n)
        {
```

Varianta PASCAL

```
procedure prim(n:integer; var
p:boolean);
var i:integer;
begin
    p:=true;
    if (n=0) and (n=1) then p:=false
        else for i:=2 to n div 2 do
            if n mod i=0 then
                p:=false;
    end;
```

Varianta PASCAL

```
VAR n,i,j:integer;
    ok:boolean;
procedure prim(n:integer; var
p:boolean);
var i:integer;
begin
    p:=true;
    if (n=0) and (n=1) then p:=false
        else for i:=2 to n div 2 do
            if n mod i=0 then
                p:=false;
    end;
BEGIN
    j:=2;
    readln(n);
    while(i<n) do
```

```

        ok=1;
        prim(j,ok);
        if(ok){ i++; cout<<j<<' ';}
        j++;
    }
}

```

Pentru neintensiv**Varianta C/C++**

```

#include<fstream.h>
ofstream fout("prime.out");
void prim(int n, int &p)
{
    int i;
    p=1;
    if (n==0||n==1) p=0;
        else
            for(i=2;i<=n/2&&p;i++)
                if (n%i==0) p=0;
}
void main()
{
    int n,j=2,ok;
    cin>>n;
    while(j<=n)
        {
            ok=1;
            prim(j,ok);
            if(ok) fout<<j<<' ';
            j++;
        }
    fout.close();
}

```

4. pentru intensiv**a)**

Ce citesc, pe rând, toate numerele din fișierul „bac.in” și se realizează vectorul de apariții ale acestor numere. Se parcurge apoi acest vector, afișându-se acei indici ai căror valori sunt nenule indicii reprezentând numerele din fișier, iar valorile din vector, frecvențele de apariție ale acestor numere.

b)**Varianta C/C++**

```

begin
    ok:=true;
    prim(j,ok);
    if (ok) then
        begin
            inc(i);
            write(j,' ');
            end;
        inc(j);
    end
end.

```

Varianta PASCAL

```

VAR n,j:integer;
    ok:boolean;
    fout:text;
procedure prim(n:integer; var
p:boolean);
var i:integer;
begin
p:=true;
if (n=0) and (n=1) then p:=false
else for i:=2 to n div 2 do
    if n mod i=0 then
p:=false;
end;
BEGIN
assign(fout,'prime.out');
rewrite(fout);
j:=2;
readln(n);
while(j<=n) do
begin
ok:=true;
prim(j,ok);
if (ok) then
write(fout,j,' ');
inc(j);
end;
close(fout);
end.

```

| Varianta PASCAL

```
#include<fstream.h>
#include<iostream.h>
int v[100],n;
ifstream fin("bac.in");
void main()
{
    int i,x;
    fin>>n;
    for(i=1;i<=n;i++)
        {fin>>x;
        v[x]=v[x]+1;
        }
    for(i=100;i>=1;i--)
        if (v[i]) cout<<i<<'
'<<v[i]<<endl;
    fin.close();
}
```

```
VAR v:array[1..100] of integer;
    n,i,x:integer;
    fin:text;
BEGIN
assign(fin,'bac.in');
reset(fin);
readln(fin,n);
for i:=1 to n do
    begin
    read(fin, x);
    v[x]:=v[x]+1;
    end;
for i:=100 downto 1 do
    if v[i]<>0 then writeln(i,'
',v[i]);
close(fin);
END.
```

pentru neintensiv

a)

Varianta C/C++

```
void max_neg(int n,int v[],int &max)
{
    int i;
    max=-10000;
    for(i=1;i<=n;i++)
        if((v[i]>max)&&(v[i]<0))
max=v[i];
    if(max===-10000) max=0;
}
```

Varianta PASCAL

```
procedure max_neg(n:integer;v:array
of integer;var max:integer);
var i:integer;
begin
max:=-10000;
for i:=1 to n do
    if (v[i]>max) and (v[i]<0) then
max:=v[i];
if max=-10000 then max:=0;
end;
```

b)

Varianta C/C++

```
#include<iostream.h>
void max_neg(int n,int v[],int &max)
{
    int i;
    max=-10000;
    for(i=1;i<=n;i++)
        if((v[i]>max)&&(v[i]<0))
max=v[i];
    if(max===-10000) max=0;
}
void main()
{
    int v[100],i,maxim,n;
    cin>>n;
    for(i=1;i<=n;i++) cin>>v[i];
    max_neg(n,v,maxim);
    if (maxim) cout<<maxim<<' ';
    else cout<<"NU EXISTA VALORI
NEGATIVE"<<endl;
    for(i=1;i<=n;i++)
        v[i]=(-1)*v[i];
```

Varianta PASCAL

```
VAR v:array[1..100] of integer;
    i,maxIM,n:integer;
procedure max_neg(n:integer;v:array
of integer;var max:integer);
var i:integer;
begin
max:=-10000;
for i:=1 to n do
    if (v[i]>max) and (v[i]<0) then
max:=v[i];
if max=-10000 then max:=0;
end;
BEGIN
rEadln(n);
for i:= 1 to n do read(v[i]);
max_neg(n,v,maxim);
if maxim<>0 then writeln(maxim,' ');
else writeln('NU EXISTA VALORI
NEGATIVE',i);
for i:=1 to n do v[i]:=(-1)*v[i];
max_neg(n,v,maxim);
```



```

max_neg(n,v,maxim);
maxim=(-1)*maxim;
if (maxim) cout<<maxim;
else cout<<"NU EXISTA VALORI
POZITIVE";
}

```

```

maxim:=(-1)*maxim;
if maxim<>0 then writeln(maxim)
else write('NU EXISTA VALORI
POZITIVE ');
end.

```

Varianta <79>

SUBIECTUL I

1. d.

2.NOTA! Varianta propusă conține o eroare în expresia logică a celei de-a doua instrucțiuni repetitive (în loc de $x|n$ trebuie scris $x|a$)

a) 12

b) Programul calculează suma divizorilor x ai numărului a . Orice număr prim are un singur divizor, în afara lui 1, pe el însuși. Exemplu: $a=23$

c)

Varianta C/C++

```

#include<iostream.h>
void main()
{
    int a,x=2,k=0,c;
    cin>>a;
    while(a>1)
    {
        c=0;
        while(a%x==0)
        {
            c=x;
            a=a/x;
        }
        if(c) k=k+x;
        x++;
    }
    cout<<k;
}

```

Varianta PASCAL

```

var a,x,k,c,n:integer;
begin
    readln(a);
    x:=2;
    k:=0;
    while (a>1) do
    begin
        c:=0;
        while (a mod x=0) do
            begin
                c:=x;
                a:=a div x;
            end;
        if c<>0 then k:=k+x;
        x:=x+1;
    end;
    write(k);
end.

```

d) citește a

$x \leftarrow 2$

$k \leftarrow 0$

┌ execută

└┬

c

0

┌ execută

c \leftarrow x

a \leftarrow [a/x]

■ cât timp $x|a$

┌ dacăc $\neq 0$ atunci

k \leftarrow k+x

■

x \leftarrow x+1

└■ cât timp $a>1$

scrie k

SUBIECTUL II

1. a

2. pentru **intensiv a**, pentru **neintensiv a**

3. [4, 0, 6, 9, 2, 5, 4, 3, 2, 6, 4, 6, 2]

4. pentru **intensiv 4**, pentru **neintensiv 5**

5.

Varianta C/C++

```
#include<iostream.h>
#include<string.h>
void main()
{
    char
sir[100],voc[]="aeiouAEIOU",*p,*q,sb[]="*";
    int i;
    cin.get(sir,100);
    for(i=0;i<strlen(voc);i++)
    { p=sir;
      while(p)
      {p= strchr(p,voc[i]);
       strcpy(q,p+1);
       strcpy(p+1,sb);
       strcat(p+2,q);
       if (p)p=p+1;
      }
    }
    cout<< sir;
}
```

Varianta PASCAL

```
var sir,voc:string;
    n,i,j:integer;
begin
readln(sir);
voc:='aeiouAEIOU';
for i:=1 to length(voc) do
    for j:=1 to length(sir) do
        if voc[i]=sir[j] then
            insert('*',sir,j+1);
writeln(sir);
end.
```

SUBIECTUL III

1. a

2. 9

3. pentru **intensiv**

Varianta C/C++

```
float max(float a, float b)
{ float v[4];
  v[1]=a; v[2]=b;
  int ia=floor(a),ib=floor(b);

  float za=a-ia,zb=b-ib;
  while(za!=(int)za)za=za*10;
  while(zb!=(int)zb)zb=zb*10;
  int i=1,j=1;
  while(ia){i=i*10;ia/=10;}
  while(ib){j=j*10;ib/=10;}
  v[3]=za+ia/i;
  v[4]=zb+ib/j;
  int maxim=v[1];
  for(i=2;i<=4;i++)
      if(v[i]>maxim) maxim=v[i];
  return maxim;
}
```

Varianta PASCAL

```
function max(a,b:real):real;
var v:array[1..4] of real;
    i,j,ia,ib:integer;
    za,zb,maxim:real;
begin
v[1]:=a; v[2]:=b;
ia:=trunc(a); ib:=trunc(b);
za:=a-ia;zb:=b-ib;
while(za<>trunc(za)) do za:=za*10;
while(zb<>trunc(zb)) do zb:=zb*10;
i:=1; j:=1;
while(ia>0)do
    begin
    i:=i*10;
    ia:=ia div 10;
    end;
while(ib>0) do
    begin
    j:=j*10;
    ib:=ib div 10;
```

Pentru neintensiv**Varianta C/C++**

```
void max_d(unsigned long int n,int
&m1,int &m2)
{
    m1=n%10;
    m2=n/10%10;
    if (m2>m1){int aux=m1;
                m1=m2;
                m2=aux;}
    n=n/100;
    while (n)
    {
        if (n%10>m1){m2=m1;m1=n%10;}
        else if (n%10>m2)m2=n%10;
        n/=10;
    }
}
```

4.**a)**

Ce citesc, pe rând, toate numerele din fișierul „**numere.in**” și se realizează vectorul de apariții ale acestor numere. Se parcurge apoi acest vector, afișându-se acei indici ai căror valori sunt nenule, iar în momentul în care un număr nu se regăsește în șir

atunci variabila logică **ok** devine falsă.

b)**Varianta C/C++**

```
end;
v[3]:=za+ia div i;
v[4]:=zb+ib div j;
maxim:=v[1];
for i:=2 to 4 do
    if v[i]>maxim then maxim:=v[i];
max:=maxim;
end;
```

Varianta PASCAL

```
procedure max_d(n:longint;VAR
m1,m2:integer);
var aux:integer;
begin
    m1:=n mod 10;
    m2:=n div 10 mod 10;
    if m2>m1 then
        begin
            aux:=m1;
            m1:=m2;
            m2:=aux;
        end;
    n:=n div 100;
    while (n>0) do
        begin
            if n mod 10 >m1 then
                begin
                    m2:=m1;
                    m1:=n mod 10;
                end
            else if n mod 10 >m2 then
                m2:=n mod 10;
            n:=n div 10;
        end;
    end;
```

| Varianta PASCAL

```
#include<fstream.h>
ifstream fin("numere.in");
int v[100],n;
void main()
{
    int i,x,ok=1;
    fin>>n;
    for(i=1;i<=n;i++) {fin>>x;
                        v[x]=1;
                    }
    for(i=1;i<=100;i++)
        if (v[i]==0) ok=0;
    if (ok) cout<<"NU LIPSESTE NICI
UN NUMAR";
        else for(i=1;i<=100;i++)
                if (v[i]==1)
cout<<i<<' ';
}
```

```
Var v:array[1..100] of integer;
    I,x,n:integer;
    ok:boolean;
fin:text;
begin
assign(fin,'numere.in'); reset(fin);
for i:=1 to n do
    begin
        read(fin,x);
        v[x]:=1;
    end;
ok:=true;
for i:=1 to n do
    if v[i]==0 then ok:=false;
if ok=true then write('NU LIPSESTE
NICI UN NUMAR') else
    for i:=1 to n do
        if v[i]==1 then write(i,' ')

END.
```

Varianta <80>

SUBIECTUL I

1. a.

2.

a) 593

b) Orice număr format doar din cifre pare. Exemplu: 13579

c)

Varianta C/C++

```
#include<iostream.h>
void main()
{
    long int a,b,p,c;
    cin>>a;
    b=0; p=1;
    while (a)
    {
        c=a%10;
        if(c%2!=0)
        {
            b=b+p*c;
            p=p*10;
        }
        a=a/10;
    }
    cout<<b;
}
```

Varianta PASCAL

```
var a,b,p,c:longint;
begin
readln(a);
b:=0;
p:=1;
while a>0 do
    begin
        c:=a mod 10;
        if c mod 2<>0 then
            begin
                b:=b+p*c;
                p:=p*10;
            end;
        a:=a div 10;
    end;
writeln(b);
end.
```

d) citeste a

$b \leftarrow 0$

$p \leftarrow 1$

┌ execută
└ $c \leftarrow a \% 10$
 └dacă $c \% 2 \neq 0$ atunci

```

    b ← b+p*c
    p ← p*10
  ■
a ← [a/10]
■cât timp a>0

```

scrie b

SUBIECTUL II

1. d

2. pentru intensiv c, pentru neintensiv d

3. 6

4. pentru intensiv 2, pentru neintensiv 5, 4, 3

5.

Varianta C/C++

```

#include<iostream.h>
#include<string.h>

void main()
{
char s[20],aux[20];
int n,k;
cin>> s;
n=strlen(s);
if (n%2==0)
{
k=n/2;
strcpy(aux,s+k);
strncat(aux,s,k);
cout<<aux;
}
else
{
k=n/2;
strcpy(aux,s+k+1);
strncat(aux,s+k,1);
strncat(aux,s,k);
cout<<aux;
}
}

```

Varianta PASCAL

```

var s,aux:string[20];
    n,k:byte;

begin
readln(s);
n:=length(s);
aux:='';
if n mod 2=0 then
    aux:=concat(copy(s,n div 2+1,n
div 2), copy(s,1,n div 2))
else
    aux:=concat(copy(s,n div 2+2,n
div 2), copy(s,n div
2+1,1),copy(s,1,n div 2));
writeln(aux);
end.

```

SUBIECTUL III

1. b

2. 11

3. pentru intensiv

Varianta C/C++

```

#include<fstream.h>
ifstream fin("fisier.in");
int v[1000],n,i,j,x,aux;
int cif_egale(int x)
{
int ok=1;
while(ok && x>10)
{
if (x%10!=x/10%10) ok=0;
x=x/10;
}
}
return ok;

```

Varianta PASCAL

```

var x,i,j,aux,n:integer;
    v:array[1..1000] of integer;
    fin:text;

function
cif_egale(x:integer):boolean;
var ok:boolean;
begin
ok:=true;
while ok and (x>10) do
begin
if x mod 10<>x div 10 mod 10
then ok:=false;

```

```

}
void main()
{
    fin>>x;
    while(!fin.eof())
        {
            if(cif_egale(x)) { n++;
v[n]=x; }
            fin>>x;
        }
    for(i=1;i<=n-1;i++)
        for(j=i+1;j<=n;j++)
            if (v[i]>v[j])
                {
                    aux=v[i];
                    v[i]=v[j];
                    v[j]=aux;
                }
}for(i=1;i<=n;i++) cout<<v[i]<<' ';

```

pentru neintensiv

Varianta C/C++

```

void patrat(int n,int v[])
{
    int i,j=0;
    for(i=1;i<=n;i++)
        {
            if (sqrt(v[i])==(int)sqrt(v[i]))
                {cout<<v[i]<<' ';
                j++;}
            if(j%10==0) cout<<endl;
        }
}

```

4. pentru intensiv

a)

Varianta C/C++

```

void cif_dis(long n, int &d)
{
    long cn;
    short k,c;

    d=1;
    while(n && d)
        {
            cn=n;

```

```

        x:=x div 10;
        end;
    cif_egale:=ok;
    end;
    begin
    assign(fin,'fisier.in');
    reset(fin);
    n:=0;
    read(fin,x);
    while (not eof(fin)) do
        begin
            if (cif_egale(x)) then
                begin
                    inc(n);
                    v[n]:=x;
                    end;
                read(fin,x);
            end;
        }
    for i:=1 to n-1 do
        for j:=i+1 to n do
            if v[i]>v[j] then
                begin
                    aux:=v[i];
                    v[i]:=v[j];
                    v[j]:=aux;
                end;
        }
    for i:=1 to n do
        write(v[i],' ');
    close(fin);
    end.

```

Varianta PASCAL

```

procedure patrat(n:integer; v:array
of integer);
var i,j:integer;
begin
    j:=0;
    for i:= 1 to n do
        begin
            if
            (sqrt(v[i])=trunc(sqrt(v[i]))) then
                begin
                    write(v[i],' ');

                    inc(j);
                    end;
                if j mod 10=0 then writeln;
            end;
        }
    end;

```

Varianta PASCAL

```

procedure cif_dis(n:longint;var
d:boolean);
var cn:longint;
    k,c:byte;
begin
    d:=true;
    while (n>0) and d do
        begin

```

```

k=1;
c=cn%10; cn=cn/10;
while(cn)
{
    if (cn%10==c) k++;
    cn/=10;
}
if (k>1) d=0;
n=n/10;
}
}

```

```

cn:=n;
k:=1;
c:=cn mod 10; cn:=cn div 10;
while (cn>0) do
begin
    if cn mod 10 = c then inc(k);
    cn:=cn div 10;
end;
if k>1 then d:=false;
n:=n div 10;
end;
end;

```

b)

Varianta C/C++

```

#include<iostream.h>
int n,i,ok;
long v[100];
void cif_dis(long n, int &d)
{
    long cn;
    short k,c;
    d=1;
    while(n && d)
    {
        cn=n;
        k=1;
        c=cn%10; cn=cn/10;
        while(cn)
        {
            if (cn%10==c) k++;
            cn/=10;
        }
        if (k>1) d=0;
        n=n/10;
    }
}
void main()
{
    cin>>n;
    for(i=1;i<=n;i++)
        cin>>v[i];
    for(i=1;i<=n;i++)
    {
        cif_dis(v[i],ok);
        if (ok) cout<<v[i]<<' ';
    }
}

```

Varianta PASCAL

```

Var v:array[1..100] of longint;
n,i:integer;
ok:boolean;
procedure cif_dis(n:longint;var
d:boolean);
var cn:longint;
k,c:byte;
begin
d:=true;
while (n>0) and d do
begin
cn:=n;
k:=1;
c:=cn mod 10; cn:=cn div 10;
while (cn>0) do
begin
if cn mod 10 = c then inc(k);
end;
if k>1 then d:=false;
n:=n div 10;
end;
end;
begin
readln(n);
for i:= 1 to n do
read(v[i]);
for i:=1 to n do
begin
cif_dis(v[i],ok);
if ok then write(v[i],' ');
end
end.

```

pentru neintensiv

a)

Varianta C/C++

```

int ordonat(long n)
{
    int x=n,ordc=1,ordd=1;
    while ((x/10) && ordc)
    {
        if(x%10<=x/10%10) ordc=0;
        x/=10;
    }
    if (!ordc)

```

Varianta PASCAL

```

function ordonat(n:longint):boolean;
var x:integer;
ordc,ordd:boolean;
begin
ordc:=true; ordd:=true;
while (x div 10 <>0) and (ordc) do
begin
if x mod 10<=x div 10 mod 10
then ordc:=false;

```

```

while ((n/10) && ordd)
{
    if(n%10>=n/10%10) ordd=0;
    n/=10;
}
if (ordc||ordd) return 1;
    else return 0;
}

```

b)

Varianta C/C++

```

#include<fstream.h>
ifstream fin("numere.in");
int ordonat(long n)
{
    int x=n,ordc=1,ordd=1;
    while ((x/10) && ordc)
    {
        if(x%10<=x/10%10) ordc=0;
        x/=10;
    }
    if (!ordc)
    while ((n/10) && ordd)
    {
        if(n%10>=n/10%10) ordd=0;
        n/=10;
    }
    if (ordc||ordd) return 1;
        else return 0;
}
void main()
{
    long n,x,i;
    fin>>n;
    for(i=1;i<=n;i++) {fin>> x;
    if(ordonat(x)) cout<<x<<' ';
        }
    fin.close();
}

```

```

x:=x div 10;
end;
if not ordc then
    while (n div 10>0) and (ordd) do
        begin
            if n mod 10 >=n div 10 mod 10 then
ordd:=false;
n:=n div 10;
end;
if (ordc) or (ordd) then
ordonat:=true
else
ordonat:=false;
end;
end;

```

Varianta PASCAL

```

VAR fin:text;
n,x,i:integer;
function ordonat(n:longint):boolean;
var x:integer;
ordc,ordd:boolean;
begin
ordc:=true; ordd:=true;
while (x div 10 <>0) and (ordc) do
begin
if x mod 10<=x div 10 mod 10
then ordc:=false;
x:=x div 10;
end;
if not ordc then
while (n div 10>0) and (ordd) do
begin
if n mod 10 >=n div 10 mod 10 then
ordd:=false;
n:=n div 10;
end;
if (ordc) or (ordd) then
ordonat:=true
else
ordonat:=false;
end;
BEGIN
assign(fin,'numere.in');
reset(fin);
readln(fin,n);
for i:=1 to n do
begin
read(fin,x);
if ordonat(x) then write(x,' ');
end;
close(fin);
END.

```


SUBIECTUL I

1. b.

2.

b) 12348

c)

Varianta C/C++

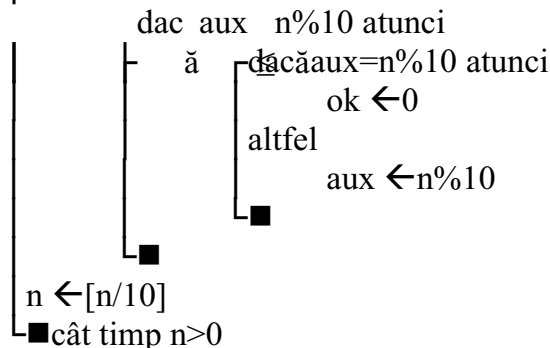
```
#include<iostream.h>
void main()
{
    int n,ok=1,aux=0;
    cin>>n;
    while(n>0)
    {
        if (aux<=n%10)
            if (aux==n%10)
                ok=0;
            else aux=n%10;
        n=n/10;
    }
    cout<<aux<<' 'ok;
```

d) citește n

ok ← 1

aux ← 0

┌ execută



scrie aux, ' ', ok

SUBIECTUL II

1. pentru **intensiv c**, pentru **neintensiv a**

2. a

3.

Varianta C/C++

a = a + b;

b = a - b;

a = a - b;

4. pentru **intensiv****Varianta C/C++**

```
p=prim;
while (p!=NULL)
{
    if(p->info%7==0) cout<<p-
```

Varianta PASCAL

```
var n,aux,ok:integer;
begin
    readln(n);
    aux:=0; ok:=1;
    while(n>0) do
        begin
            if (aux<=n mod 10) then
                if (aux=n mod 10) then
                    ok:=0;
                else aux:=n mod 10;
            n:=n div 10;
        end;
    writeln(aux,' ',ok);
End.
```

Varianta PASCAL

a:=a+b;

b:=a-b;

a:=a-b;

Varianta PASCAL

```
p:=prim;
while p<>nil do
begin
    if (p^.info mod 7 = 0) then
```

```
>info<<' \';
    p=p->adr;
}
```

Pentru neintensiv

Varianta C/C++

```
#include<iostream.h>
#include<string.h>
void main()
{
    char sir[25];
    int n;
    cin>>sir;
    n=strlen(sir);
    if(n%2) cout<<sir[n/2];
        else cout<<sir[n/2-
1]<<sir[n/2];
}
```

5. pentru intensiv

Varianta C/C++

```
#include<iostream.h>
void main()
{
    int k,a[10][10];
    short int n,m,i,j,s=0;
    cin>>k>>n>>m;
    for(i=1;i<=n;i++)
        for(j=1;j<=m;j++)
            cin>>a[i][j];
    for(j=1;j<=m;j++){
        short int gasit=0;
        for(i=1;i<=n;i++)
            if(a[i][j]==k)
                gasit=1;
        if (gasit) s+=j;
    }
    cout<<s;
}
```

Pentru neintensiv

Varianta C/C++

```
#include<iostream.h>
void main()
{int
n,m,i,j,a[10][10],min,imin,jmin,max
,imax,jmax;
cin>>n>>m;
for(i=1;i<=n;i++)
    for(j=1;j<=m;j++)
        cin>>a[i][j];
min=max=a[1][1];
imin=imax=jmin=jmax=1;
for(i=1;i<=n;i++)
    for(j=1;j<=m;j++)
        if (a[i][j]<min)
{min=a[i][j]; imin=i; jmin=j;}
else if
(a[i][j]>max){max=a[i][j];
imax=i;jmax=j;}
```

```
write(p^.info, '\ ');
p:=p^.adr;
end;
```

Varianta PASCAL

```
VAR sir:string[25];
    n:integer;
begin
readln(sir);
n:=length(sir);
if n mod 2 = 1 then write(sir[n div
2+1])
    else write(sir[n div 2],sir[n div
2 +1]);
end.
```

Varianta PASCAL

```
var a:array[1..10,1..10] of integer;
    k,n,m,i,j,s:integer;
    gasit:boolean;
begin
readln(k,n,m);
for i:=1 to n do
    for j:=1 to m do
        read(a[i,j]);
s:=0;
for j:=1 to m do
    begin
    gasit:=false;
    for i:=1 to n do
        if a[i,j]=k then gasit:=true;
    if gasit then s:=s+j;
    end;
writeln(s);
end.
```

Varianta PASCAL

```
VAR
n,m,i,j,min,imin,jmin,max,imax,jmax,a
ux:integer;
a:array[1..10,1..10] of integer;
begin
readln(n,m);
for i:=1 to n do
    for j:=1 to m do
        read(a[i,j]);
min:=a[1,1]; imin:=1; jmin:=1;
max:=a[1,1]; imax:=1; jmax:=1;
for i:=1 to n do
    for j:=1 to m do
        if a[i,j]<min then
            begin
jmin:=j;
min:=a[i,j]; imin:=i;
end
        else if a[i,j]>max then
```

```
int aux=a[imin][jmin];
  a[imin][jmin]=a[imax][jmax];
  a[imax][jmax]=aux;
  for(i=1;i<=n;i++)
    { for(j=1;j<=m;j++)
      cout<<a[i][j]<<' ';
      cout<<endl;
    }
}
```

```
begin
max:=a[i,j];
imax:=i;
jmax:=j;
end;
aux:=a[imin,jmin];
a[imin,jmin]:=a[imax,jmax];
a[imax,jmax]:=aux;
for i:=1 to n do
begin
for j:=1 to m do
write(a[i,j], ' ');
writeln;
end;
end.
```

SUBIECTUL III

1. b

2. pentru **intensiv 3**, pentru **neintensiv 4**

3.

Pentru intensiv

Varianta C/C++

```
float medie(int n,int a[])
{
int i,s=0;
for(i=1;i<=n;i++)
s+=a[i];
return s/n;
}
```

Pentru neintensiv

a)

```
int ordonat(int n,int v[],int k1,
int k2)
```

b)

```
#include<iostream.h>
int ordonat(int n,int v[],int k1,
int k2)
```

```
{
int i,ok=1;
for(i=k1;i<=k2&&ok;i++)
if(v[i]>v[i+1]) ok=0;
return ok;
}
```

```
void main()
```

```
{
int v[100],n,i;
cin>>n;
for(i=1;i<=n;i++)
cin>>v[i];
int j,kmax=0,k,cj;
i=1;
while(i<=n)
{
j=i+1; k=1;
while(ordonat(n,v,i,j))
{j++;k++;}
```

Varianta PASCAL

```
function medie(n:integer;a:array of
integer):real;
```

```
var i,s:integer;
```

```
begin
```

```
s:=0;
```

```
for i:=0 to n-1 do
```

```
s:=s+a[i];
```

```
medie:=s/n;
```

```
end;
```

a)

```
function ordonat( n,int v[],int k1,
int k2)
```

```

    if(k>kmax) {kmax=k; cj=j;}
    i=i+j-1;
    }
    cout<<cj-kmax<<' ' <<cj;
}

```

4. Pentru intensiv

a)

Varianta C/C++

```

#include<fstream.h>
ifstream fin("numere.in");
ofstream fout("numere.out");
void main()
{
    int v[500],n,i,x;
    for(i=0;i<=500;i++) v[i]=0;
    while(!fin.eof())
    { fin>>x;
      v[x]=v[x] +1; }
    for(i=1;i<=500;i++)
        if(v[i]==1) fout<<i<<' \';
    fin.close(); fout.close();
}

```

Pentru neintensiv

Varianta C/C++

```

#include<fstream.h>
ifstream fin("numere.in");
void main()
{
    int x,y;

    do
        fin>>x>>y;
        if(x>y) cout<<x<<endl;
        else cout<<y<<endl;
    } while (!fin.eof());
    fin.close();
}

```

Varianta PASCAL

```

var fin,fout:text;
    v:array[1..500] of integer;
    n,i,x:integer;
begin
    assign(fin,'numere.in'); reset(fin);
    assign(fout,'numere.out');rewrite(fout);
    while(not eof(fin)) do
    begin
        read(fin,x);
        v[x]:=v[x]+1;
    end;
    for i:=1 to 500 do
        if (v[i]=1) write(fout,i,' ');
    close(fin); close(fout);
end.

```

Varianta PASCAL

```

VAR fin:text;
    x,y:integer;
begin
    assign(fin,'numere1.in');reset(fin);
    repeat
        readln(fin,x,y);
        if x>y then writeln(x)
            else writeln(y);
    until eof(fin);
    close(fin);
end.

```

b) **pentru intensiv** Se folosește un vector de frecvențe în care se memorează numărul de apariții al numerelor citite din fisierul “numere.in”. Parcurgem vectorul

construit și afișăm indicii valorilor egale cu 1 (care au apărut o singură dată în

Varianta <82>

SUBIECTUL I

1. d.

2.

a) 2515

b) m=7

Cea mai mare valoare a cifrei m trebuie să fie conținută în numărul afișat (761)

c)

Varianta C/C++

```

#include<iostream.h>
void main()
{

```

Varianta PASCAL

```

var m,n,i,x,aux,ok:integer;
begin
    read(m); read(n);

```

```

unsigned int m,n;
int i,x,aux,ok;
cin>>m;
cin>>n;
for (i=1;i<=n;i++)
    {
    cin>>x;
    aux=x;
    ok=0;
    while (x>0) {
        if (x%10==m) ok=1;
        x=x/10;
    }
    if (ok==1) cout<<aux;
    }
}

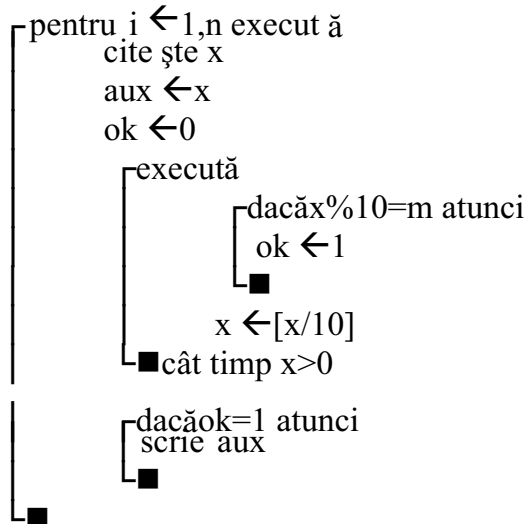
```

```

for i:=1 to n do
begin
    read(x);
    aux:=x;
    ok:=0;
    while (x>0) do begin
        if x mod 10=n
        then ok:=1;
        x:=x div 10;
    end;
    if ok=1 then write(aux);
end;
end.

```

d) citește m
citește n



SUBIECTUL II

1. b

2. pentru **intensiv a**, pentru **neintensiv c**

NOTĂ! Răspunsurile sunt numerotate greșit la varianta pentru neintensiv. Răspunsul corect este **g**, echivalentul lui **c**.

3. “tarta”

4. pentru intensiv

Varianta C/C++

```

p=prim;
while (p!=NULL)
    {
        if(p->info%10==0) cout<<p-
>info<<' \';
        p=p->adr;
    }

```

Varianta PASCAL

```

p:=prim;
while p<>nil do
begin
    if (p^.info mod 10 = 0) then
write(p^.info, ' \');
p:=p^.adr;
end;

```

pentru neintensiv

Numărul de noduri cu grad par este: **2**, iar numărul de noduri cu grad impar este: **2**

5.

Varianta C/C++

```
#include<iostream.h>
void main()
{
    int k,a[10][10];j,p=1;
    cin>>k>>n>>m;
    for(i=1;i<=n;i++)
        for(j=1;j<=m;j++)
            cin>>a[i][j];
    for(j=1;j<=m;j++){
        short int gasit=0;
        for(i=1;i<=n;i++)
            if(a[i][j]==k)
                gasit=1;
        if (gasit) p*=j;
    }
    cout<<p;
}
```

pentru neintensiv

Varianta C/C++

```
#include<iostream.h>
void main()
{
    int
    n,i,j,a[10][10],min,imin,max,imax;
    cin>>n;
    for(i=1;i<=n;i++)
        for(j=1;j<=n;j++)
            cin>>a[i][j];
    min=max=a[1][1];
    imin=imax=1;
    for(i=1;i<=n;i++)
        if (a[i][i]<min)
            {min=a[i][i]; imin=i;}
        else if
            (a[i][i]>max) {max=a[i][i];
            imax=i;}
    int aux=a[imin][imin];
    a[imin][imin]=a[imax][imax];
    a[imax][imax]=aux;
    for(i=1;i<=n;i++)
        { for(j=1;j<=n;j++)
            cout<<a[i][j]<<' ';
            cout<<endl;
        }
}
```

Varianta PASCAL

```
var a:array[1..10,1..10] of integer;
    k,n,m,i,j,p:integer;
    gasit:boolean;

begin
    readln(k,n,m);
    for i:=1 to n do
        for j:=1 to m do
            read(a[i,j]);
    p:=1;
    for j:=1 to m do
        begin
            gasit:=false;
            for i:=1 to n do
                if a[i,j]=k then gasit:=true;
            if gasit then p:=p*j;
        end;
    writeln(p);
end.
```

Varianta PASCAL

```
VAR
n,m,i,j,min,imin,max,imax,aux:integer;
a:array[1..10,1..10] of integer;
begin
    readln(n);
    for i:=1 to n do
        for j:=1 to n do
            read(a[i,j]);
    min:=a[1,1];imin:=1;
    max:=a[1,1];imax:=1;
    for i:=1 to n do
        if a[i,i]<min then
            begin
                min:=a[i,i]; imin:=i;
            end
        else if a[i,i]>max then
            begin
                max:=a[i,i];
                imax:=i;
            end;
    aux:=a[imin,imin];
    a[imin,imin]:=a[imax,imax];
    a[imax,imax]:=aux;
    for i:=1 to n do
        begin
            for j:=1 to n do
                write(a[i,j], ' ');
            writeln;
        end;
end.
```

SUBIECTUL III

1. pentru intensiv b, pentru neintensiv b

2. 5

3.

Pentru intensiv

Varianta C/C++

```
int suma(int n,int a[])
{
    int i,s=0,uc,puc;
    for(i=1;i<=n;i++)
    {
        uc=a[i]%10;
        puc=a[i]/10%10;
        if (puc==uc) s+=a[i];
    }
    return s;
}
```

Pentru neintensiv

Varianta C/C++

```
float medie(int n,int a[])
{
    int i,s=0,k=0;
    for(i=1;i<=n;i++)
        if (a[i]%2)
            s+=a[i];
    k++;
    if (k) return s/k;
    else return 0;
}
```

4. Pentru intensiv

a)

Varianta C/C++

```
#include<fstream.h>
ifstream fin("numere.in");
ofstream fout("numere.out");
void main()
{
    int v[500],n,i,x;
    for(i=0;i<=500;i++) v[i]=0;

    while(!fin.eof())
    {
        fin>>x;
        v[x]=v[x]+1;
        i=99;
        while(v[i]==0) i--;
        fout<<i<<' '<<v[i]<<endl;
        i=10;
        while(v[i]==0) i++;
        fout<<i<<' '<<v[i]<<endl;
        fin.close(); fout.close();
    }
}
```

Varianta PASCAL

```
function suma(n:integer;a:array of
integer):integer;
var uc,puc,i,s:integer;
begin
    s:=0;
    for i:=0 to n-1 do
        begin
            uc:=a[i] mod 10;
            puc:=a[i] div 10 mod 10;
            if (puc=uc) then s:=s+a[i];
        end;
    suma:=s;
end;
```

Varianta PASCAL

```
function medie(n:integer;a:array of
integer):real;
var k,i,s:integer;
begin
    s:=0;k:=0;
    for i:=0 to n-1 do
        if (a[i] mod 2=1) then
            begin
                s:=s+a[i];
                inc(k);
            end;
    if (k>0) then medie:=s/k;
    else medie:=0;
end;
```

Varianta PASCAL

```
var fin,fout:text;
    v:array[1..500] of integer;
    n,i,x:integer;
begin
    assign(fin,'numere.in'); reset(fin);
    assign(fout,'numere.out');rewrite(fout);
    while(not eof(fin)) do
        begin
            read(fin,x);
            v[x]:=v[x]+1;
        end;
    i:=99;
    while(v[i]=0) do dec(i);
    writeln(fout,i,' ',v[i]);
    i:=10;
    while(v[i]=0) do inc(i);
    writeln(fout,i,' ',v[i]);
    close(fin); close(fout);
end.
```

b) Se folosește un vector de frecvențe în care se memorează numărul de apariții al numerelor citite din fisierul “numere.in”. Pentru găsirea celui mai mic, respectiv a celui mai

mare număr de două cifre se caută prima poziție nenulă din vector între 10 și 99 în sens crescător, respectiv descrescător.

Pentru neintensiv

a)

Varianta C/C++

```
#include<fstream.h>
ifstream fin("numere.in");
void main()
{
    int v[500],n,i,x;
    for(i=0;i<=500;i++) v[i]=0;
    do
    { fin>>x;
      v[x]=v[x] +1;
    }while(!fin.eof());
    i=1;
    while(i<=500)
    {
        if (v[i]==1) cout<<i<<' ';
        i++;
    }
    fin.close();
}
```

Varianta PASCAL

```
var fin:text;
    v:array[1..500] of integer;
    n,i,x:integer;
begin
    assign(fin,'numere.in'); reset(fin);
    repeat
        read(fin,x);
        v[x]:=v[x]+1;
    until eof(fin);
    i:=1;
    while(i<=500) do
        begin
            if v[i]=1 then write(i,' ');
            inc(i);
        end;
    close(fin);
end.
```

b) Se folosește un vector de frecvențe în care se memorează numărul de apariții al numerelor citite din fisierul “numere.in”. Se parcurge în sens crescător acest vector construit afișând acei indici pentru care valorile sunt egale cu 1 (elementele apar o singură dată).

Varianta <83>

SUBIECTUL I

1. a.

2.

a) “nu”

b) 999

c)

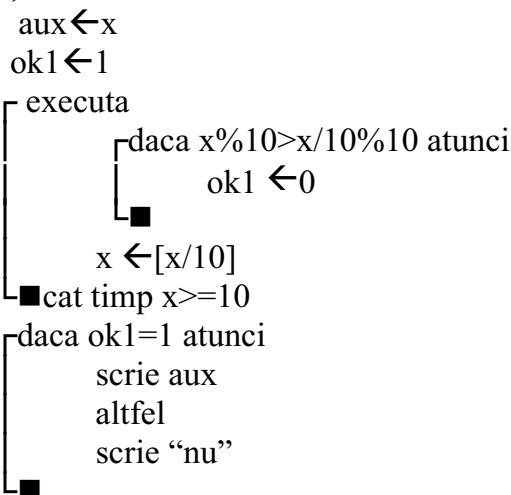
Varianta C/C++

```
#include<iostream.h>
void main()
{
    unsigned int x;
    int aux,ok1;
    cin>>x;
    aux=x;
    ok1=1;
    while (x>=10) {
        if (x%10>x/10%10)
            ok1=0;
        x=x/10;
    }
    if (ok1==1) cout<<aux;
    else cout<<"nu";
}
```

Varianta PASCAL

```
var x,aux,ok1:integer;
begin
    read(x);
    aux:=x;
    ok1:=1;
    while (x>=10) do
        begin
            if x mod 10>x div 10 mod 10
            then ok1:=0;
            x:=x div 10;
        end;
    if ok1=1 then write(aux)
    else write('nu');
end.
```


d) citește x



SUBIECTUL II

1. pentru **intensiv c**, pentru **neintensiv 2**
2. pentru **intensiv b**, pentru **neintensiv a**
3. Pentru **intensiv**

Graful este format din doua subgrafuri ce au câte 3 noduri și 3 muchii fiecare.

```

0 1 1 0 0 0
1 0 1 0 0 0
1 1 0 0 0 0
0 0 0 0 1 1
0 0 0 1 0 1
0 0 0 1 1 0
  
```

Pentru **neintensiv**

```

0 1 1 0 0
1 0 1 0 0
1 1 0 0 0
0 0 0 0 1
0 0 0 1 0
  
```

4.

Varianta C/C++

```

short int a[10][10], n, m, i, j, k;

for (i=1; i<=m; i++)
    for (j=1; j<=n; j++)
        cin >> a[i][j];

cin >> k;
for (i=1; i<=m; i++)
    if (a[k][i] % 2 == 1)
        cout << a[k][i] << ' ';
  
```

Pentru **neintensiv**

Varianta C/C++

```

#include <iostream.h>
#include <string.h>

void main()
{
    char sir[25], voc[] = "aeiou";
    cin >> sir;
  
```

Varianta PASCAL

```

Var a:array[1..10,1..10] of byte;

readln(n,m); k:integer;
for i:=1 to n do
    for j:= 1 to m do
        read(a[i,j]);
readln(k);
for i:=1 to m do
    if (a[k,j] mod 2=1) write(a[k,j],
    ');
  
```

Varianta PASCAL

```

VAR sir,voc:string[25];
    i,j:integer;

BEGIN
    readln(sir);
    voc:='aeiou';
    for i:=1 to length(sir) do
  
```

```
int i,j;
for(i=0;i<strlen(sir);i++)
    for(j=0;j<strlen(voc);j++)
        if (sir[i]==voc[j])
cout<<sir[i];
}
```

5. Pentru intensiv

Varianta C/C++

```
struct lista { char info;
              lista
*urm,*prec;};
lista *prim,*sf;
char aux;
lista *p=prim,*q=ultim,*r=prim;
while (r!=NULL)
{aux=p->info;
 p->info=q->info;
 q->info=aux;

 p=p->urm; q=q->prec;
 r=r->urm->urm;
}
```

Pentru neintensiv

Varianta C/C++

```
#include<iostream.h>
void main()
{
    int
n,i,j,a[10][10],min,imin,max,imax;
    cin>>n;
    for(i=1;i<=n;i++)
        for(j=1;j<=n;j++)
            cin>>a[i][j];
    min=max=a[1][n];
    imin=imax=1;
    for(i=1;i<=n;i++)
        if (a[i][n-i+1]<min)
{min=a[i][n-i+1]; imin=i; }
        else if (a[i][n-
i+1]>max){max=a[i][n-i+1];
imax=i;}
    int aux=a[imin][n-imin+1];
    a[imin][n-imin+1]=a[imax][n-
imax+1];
    a[imax][n-imax+1]=aux;
    for(i=1;i<=n;i++)
    { for(j=1;j<=n;j++)
        cout<<a[i][j]<<' ';
        cout<<endl;
    }
}
```

```
for j:=1 to length(voc) do
    if sir[i]=voc[j] then
write(sir[i]);
END.
```

Varianta PASCAL

```
type lista=^nod;
nod=record
    info:char;
    urm,prec:lista;
end;
Var prim, sf,p,q,r:lista;
aux:char;
p=prim; q=ultim; r=prim;
while(r<>nil)
begin
    aux:=p^.info;
    p^.info:=q^.info;
    q^.info:=aux;
    p=p^.urm; q:=q^.prec;
    r:=r^.urm^.urm;
end;
```

Varianta PASCAL

```
VAR
n,m,i,j,min,imin,max,imax,aux:integer;
a:array[1..10,1..10] of integer;
begin
readln(n);
for i:=1 to n do
    for j:=1 to n do
        read(a[i,j]);
min:=a[1,n];imin:=1;
max:=a[1,n];imax:=1;
for i:=1 to n do
    if a[i,n-i+1]<min then
        begin
            min:=a[i,n-i+1]; imin:=i;
        end
    else if a[i,n-i+1]>max then
        begin
            max:=a[i,n-i+1];
            imax:=i;
        end;
aux:=a[imin,n-imin+1];
a[imin,n-imin+1]:=a[imax,n-imax+1];
a[imax,n-imax+1]:=aux;
for i:=1 to n do
    begin
        for j:=1 to n do
            write(a[i,j], ' ');
        writeln;
    end;
end.
```

SUBIECTUL III

1. pentru intensiv b, pentru neintensiv c

2. 11

3.

Pentru intensiv

Varianta C/C++

```
int suma(int n,int a[])
{
    int i,s=0,uc,pc;
    for(i=1;i<=n;i++)
        {
            uc=a[i]%10;
            pc=a[i];
            while (pc>9)
                pc=pc/10;
            if (pc==uc) s+=a[i];
        }
    return s;
}
```

Pentru neintensiv

Varianta C/C++

```
#include<iostream.h>
float medie(int n,int a[])
{
    int i,s=0,k=0;
    for(i=1;i<=n;i++)
        if (a[i]%2==0) { s+=a[i];
        k++; }
    if (k) return s/k;
    else return 0;
}
void main()
{
    int v[100],n,i;
    cin>>n;
    for(i=1;i<=n;i++) cin>>v[i];
    cout<<medie(n,v);
}
```

4. pentru intensiv

a)

Varianta C/C++

```
#include<fstream.h>
ifstream fin("numere3.in");
void main()
{
    int x,i,k,kmin=32000,max,maxf;
    while(!fin.eof())
        {
            fin>>x; max=x; k=0;
            while(x)
                {
```

Varianta PASCAL

```
function suma(n:integer;a:array of
integer):integer;
var uc,pc,i,s:integer;
begin
    s:=0;
    for i:=0 to n do
        begin
            uc:=a[i] mod 10;
            pc:=a[i];
            while (pc>9) do pc:= pc div 10;
            if (pc=uc) then s:=s+a[i];
        end;
    suma:=s;
end;
```

Varianta PASCAL

```
var v:array[0..100] of integer;
    n,i:integer;
function medie(n:integer;a:array of
integer):real;
var k,i,s:integer;
begin
    s:=0;k:=0;
    for i:=0 to n-1 do
        if (a[i] mod 2=0) then
            begin
                s:=s+a[i];
                inc(k);
            end;
    if (k>0) then medie:=s/k
    else medie:=0;
end;
begin
    readln(n);
    for i:=0 to n-1 do
        read(v[i]);
    writeln(medie(n,v));
end.
```

Varianta PASCAL

```
var fin:text;
    x,i,k,max,kmin,maxf:integer;
begin
    assign(fin,'numere.in');
    reset(fin);
    kmin:=32000;
    while (not eof(fin)) do
        begin
            read(fin,x);max:=x;k:=0;
            while(x<>0) do
```

```

        fin>>x; k++;
        if(x>max) max=x;
    }
    if (k<kmin)
        { maxf=max; kmin=k;}
    }
cout<<maxf;
fin.close();
}

```

```

begin
read(fin,x);inc(k);
if (x>max) then max:=x;
end;
if (k<kmin) then
begin maxf:=max;kmin:=k; end;
end;
writeln(maxf);
close(fin);
end.

```

b) Se folosește o singură variabilă pentru citirea numerelor din fișier. Se calculează numărul de elemente ale fiecărei linii și maximul dintre acestea. Elementul maxim al unei linii este comparat cu elementul maxim al liniei următoare, reținând valoarea cea mai mare pentru linia cu număr mai mic de elemente.

4. pentru neintensiv

a)

Varianta C/C++

```

#include<fstream.h>
ifstream fin("numere5.in");
void main()
{
    int v[500],n,i,x;
    for(i=1;i<=500;i++) v[i]=0;
    do
    { fin>>x;
      v[x]=v[x] +1;
    }while(!fin.eof());
    i=500;
    while(i>=1)
    {
        if (v[i]==1) cout<<i<<' ';
        i--;
    }
    fin.close();
}

```

Varianta PASCAL

```

var fin:text;
    v:array[1..500] of integer;
    n,i,x:integer;
begin
assign(fin,'numere5.in');
reset(fin);
repeat
read(fin,x);
v[x]:=v[x]+1;
until eof(fin);
i:=500;
while(i>=1) do
begin
if v[i]=1 then write(i,' ');
dec(i);
end;
close(fin);
end.

```

b) Se folosește un vector de frecvențe în care se memorează numărul de apariții al numerelor citite din fișierul “numere.in”. Se parcurge în sens descrescător acest vector construit afișând acei indici pentru care valorile sunt egale cu 1 (elementele apar o singură dată).

Varianta <84>

SUBIECTUL I

1. c.

2.

a) 6

b) 968

c)

Varianta C/C++

```

#include<iostream.h>
void main()
{
    int n,ok=0,ok1,c;
    cin>>n;

```

Varianta PASCAL

```

var n,ok,ok1,c:integer;
begin
readln(n);
ok:=0;
while(n>0) do

```

```

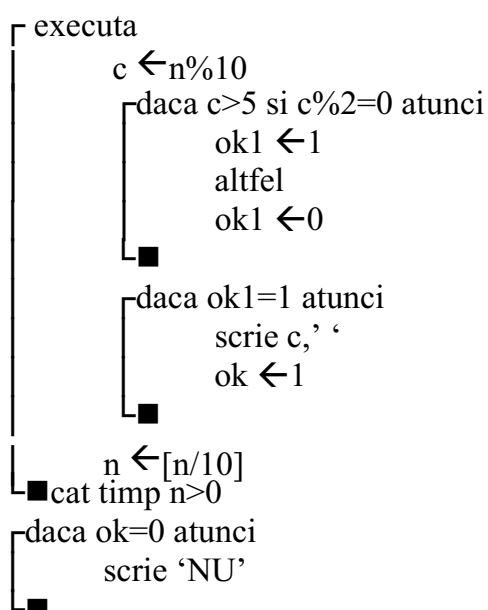
while (n>0)
{
    c=n%10;
    if (c>5 && c%2==0) ok1=1;
        else ok1=0;
    if (ok1==1)
        {
            cout<<c<<' ';
            ok=1;
        }
    n/=10;
}
if (!ok) cout<<"NU";
}
    
```

```

begin
    c:=n mod 10;
    if (c>5) and (c mod 2=0) then
        ok1:=1
    else
        ok1:=0;
    if (ok1=1) then
        begin
            write(c,' ');
            ok:=1;
        end;
    n:=n div 10;
end;
if ok=0 then writeln('NU');
end.
    
```

d) citeste n

ok ← 0



SUBIECTUL II

1. pentru **intensiv d**, pentru **neintensiv a**

2. pentru **intensiv a**, pentru **neintensiv b**

3.

Varianta C/C++

```

char c;
cin>>c;
if (c!='z' && c!='Z')
cout<<(char)(c+1);
    
```

Varianta PASCAL

```

var c:char;
readln(c);
if (c<>'z') and (c<>'Z') then
    writeln(chr(ord(c)+1));
    
```

4. pentru **intensiv**

Varianta C/C++

```

int a[10][10], p; i, j;
for (i=1; i<=n; i++)
    for (j=1; j<=n; j++)
        if (a[i][j]%2==0) p*=a[i][j];
cout<<p;
    
```

Varianta PASCAL

```

Var a:array[1..10,1..10] of integer;
p=1;
for i:=1 to n do
    for j:=1 to n do
        if a[i][j] mod 2 = 0 then
            p:=p*a[i][j];
    
```

pentru neintensiv

Varianta C/C++

```
cout<<el. nume<<el.prenume<<el.mediabac;
```

5. pentru intensiv

Varianta C/C++

```
struct lista { unsigned int info; lista
*urm; };
lista *prim,*sf;
lista *p=prim;
while (p->urm!=NULL) p=p->urm;
cout<<(prim->info*p->info);
```

pentru neintensiv

Varianta C/C++

```
#include<iostream.h>
void main()
{
int a[10][10],n,i,j,p=1;
cin>>n;
for(i=1;i<=n;i++)
for(j=1;j<=n;j++)
cin>>a[i][j];

for(i=1;i<=n;i++)
if(a[i][i]%2==0) p*=a[i][i];
if(p==1) cout<<"imposibil";
else cout<<p;
}
```

SUBIECTUL III

1. pentru intensiv d, pentru neintensiv d

2. pentru intensiv 101, pentru neintensiv 101

3. pentru intensiv

a)

Varianta C/C++

```
#include<iostream.h>
void main()
{
int
n,a[100],b[100],i,j,sib=0,spa=0;
cin>>n;
for(i=1;i<=n;i++) cin>>a[i];
for(i=1;i<=n;i++) cin>>b[i];
for(i=1;i<=n;i++) if (a[i]%2==0)
spa=spa+a[i];
for(i=1;i<=n;i++) if (b[i]%2 &&
b[i]<spa) sib=sib+b[i];
cout<<sib;
}
```

286

```
if (a[i,i] mod 2=0) then
p:=p*a[i,i];
writeln(p);
```

Varianta PASCAL

```
writeln(el.nume,el.prenume,el.mediaba
c);
```

Varianta PASCAL

```
type lista=^nod;
nod=record
info:integer;
urm:lista;
end;
Var prim, sf,p:lista;
p=prim;
while(p^.urm<>nil) do p:=p^.urm;
writeln(prim^.info*p^.info);
```

Varianta PASCAL

```
Var a:array[1..10,1..10] of integer;
n,i,j,p:integer;
begin
readln(n);
p:=1;
for i:=1 to n do
for j:=1 to n do
read(a[i,j]);
for i:=1 to n do
if (a[i,i] mod 2=0); then
if (p=1) then
writeln('Imposibil')
else
writeln(p);
end.
```

Pentru neintensiv

Varianta C/C++

```
#include<iostream.h>
void main()
{
    int a[100],b[100],c[100],i,n;
    cin>>n;
    for(i=1;i<=n;i++)
        cin>>a[i];
    for(i=1;i<=n;i++)
        cin>>b[i];
    for(i=1;i<=n;i++)
        c[i]=a[i]+b[i];
    for(i=1;i<=n;i++)
        cout<<c[i]<<' ';
}
```

Varianta PASCAL

```
VAR a,b,c:array[1..100]of integer;
    n,i,j,sib,spa:integer;
BEGIN
    readln(n);
    for i:=1 to n do read(a[i]);
    for i:=1 to n do read(b[i]);
    for i:=1 to n do
        c[i]:=a[i]+b[i];
    for i:=1 to n do
        write(c[i],' ');
END.
```

NOTĂ! Ultimul element al sirului c este 16, nu 9 pentru exemplul din varianta.

4.

a) int cmmmc(int a,int b)

b)

Varianta C/C++

```
#include<fstream.h>
ifstream fin("numere2.in") ;
ofstream fout("numere2.out");
int cmmmc(int a,int b)
{
    int ca=a,cb=b;
    while(ca!=cb)
        if (ca>cb) ca=ca-cb;
        else cb=cb-ca;
    return (a*b/ca);
}
void main()
{
    int x,y;
    while(!fin.eof())
    {
        fin>>x>>y;
        cout<<cmmmc(x,y)<<endl;
    }
    fin.close();
    fout.close();
}
```

a)function cmmmc(a,b:integer):integer;

b)

Varianta PASCAL

```
var fin,fout:text;
    x,y:integer;
function
cmmmc(a,b:integer):integer;
var ca,cb:integer;
begin
    ca:=a; cb:=b;
    while(ca<>cb) do
        if (ca>cb) then ca:=ca-cb
        else cb:=cb-ca;
    cmmmc:=(a*b) div ca;
end;
begin
    assign(fin,'numere2.in');
    reset(fin);
    assign(fout,'numere2.out');
    rewrite(fout);
    while (not eof(fin)) do
        begin
            readln(fin,x,y);
            writeln(fout,cmmmc(x,y));
        end;
    close(fin); close(fout);
end.
```

Varianta <85>

SUBIECTUL I

1. b

2.

a) 5 9

b) 879

Varianta C/C++

```
#include<iostream.h>
void main()
```

Varianta PASCAL

```
program p1;
var n,c:word;
```

```
{unsigned n, c;
 int ok, ok1;
 do{cout<<"n="; cin>>n;}
 while(n<=1);
 while(n>0)
 {c=n%10;
  if(c%2==1) ok1=1;
  else ok1=0;
  if(ok1==1) {cout<<c<<' ';ok=1;}
  n/=10;
 }
 if(ok==0) cout<<"nu";
 }
```

```
ok, ok1:byte;
begin
repeat write('n=');read(n);until
n>1;
while n>0 do
begin c:=n mod 10;
if c mod 2 =0 then ok1:=1
else ok1:=0;
if ok1 =1 then
begin write(c, ' '); ok:=1; end;
n:=n div 10;
end;
if ok=0 then write('nu');
end.
```

d) citește n

ok ← 0

repetăc ← n%10

dacă c%2=1 atunci ok1 ← 1
altfel ok1 ← 0

dacă ok1=1 atunci

scrie c, ' '

ok ← 1

n ← [n/10]

până când n=0

dacă ok=0 atunci scrie "nu"

ȘUBIECTUL II

1. a

2. b

3.

Varianta C/C++

Intensiv

```
char s[31]; int n;
cin.get(s, 31);
n=strlen(s);
cout<<s[0]<<s[n-1];
```

Neintensiv

```
cout<<"nume:";cin>>p.nume;
cout<<"prenume:";cin>>p.prenume;
cout<<"salariu:";cin>>p.salariu;
```

4.

Varianta C/C++

```
#include<iostream.h>
void main()
{unsigned a[4][4], p;
 //citirea matricei
 p=a[0][0]*a[1][1]*a[2][2]*a[3][3];
 cout<<p;}
```

Varianta PASCAL

Intensiv

```
var s:string[30];
begin
readln(s);
writeln(s[1], ' ', s[length(s)]);
end.
```

Neintensiv

```
write("nume:");readln(p.nume);
write("prenume:");readln(p.prenume);
write("salariu:");readln(p.salariu);
```

Varianta PASCAL

program p3;

var a:array[1..4,1..4] of byte;

p:word;

begin {citirea matricei}

p:=a[1,1]*a[2,2]*a[3,3]*a[4,4];

write(p);

end.

5. Vom afișa nodurile q cu proprietatea căsqrt(q->info) este număr întreg.

Varianta C/C++

```
#include<iostream.h>
struct nod{unsigned info;
           nod *adr;} *prim, *q;
void main()
{
    //crearea listei
    q=prim;
    while(q)
        {if(sqrt(q->info)==int(sqrt(q->info))
          cout<<q->info<<' ';
          q=q->adr;}}
```

Varianta PASCAL

```
program p4;
type lista=^nod;
      nod=record info:word;
                adr:lista;
      end;
var p,q:lista;
begin{crearea listei}
  q:=p;
  while q<>nil do begin
    if
      sqrt(q^.info)=trunc(sqrt(q^.info))
      then write(q^.info,' ');
    q:=q^.adr
  end;end.
```

SUBIECTUL III

1. a

3. Intensiv 10, Neintensiv 35: Pe parcursul citirii valorilor primului șir calculăm media elementelor pare, iar pe parcursul citirii celui de al doilea, numărăm valorile mai mici decât media determinată.

Varianta C/C++

```
#include<iostream>
void main()
{unsigned n,i,c=0,s;
 int a,b;
 float m=0;
 cout<<"n=";cin>>n;
 for(i=1;i<=n;i++)
   {cout<<"a["<<i<<"]=";cin>>a;
   }if(a%2==0){m+=a;c++;}
 if(c>0)
   {m/=c;s=0;
   for(i=1;i<=n;i++)
     {cout<<"b["<<i<<"]=";cin>>b;
     if(b<m)s+=b;}
   cout<<"s="<<s; }
 else cout<<"sirul a nu contine
 valori pare"; }
```

Varianta PASCAL

```
program p4;
var n,i,c,s:word;
    a,b:integer;
    m:real;
begin
  write('n=');read(n);
  for i:=1 to n do
    begin write('a[' ,i, ']=' );read(a);
    m:=m+a;inc(c);end;
  end;
  if c>0 then
    begin m:=m / c;s:=0;
    for i:=1 to n do
      begin
        write('b[' ,i, ']=' );read(b);
        if b<m then s:=s+b; end;
        write('s=',s); end
    else write('sirul a nu contine
valori pare'); end.
```

4. a)

Varianta C/C++

```
unsigned cmmdc(unsigned a, unsigned
b);
```

b)

```
#include<iostream.h>
#include<fstream.h>
unsigned cmmdc(unsigned a, unsigned
b);
void main()
{
  unsigned x,y,d;
  ifstream f("numere.in");
  ofstream g("numere.out");
  while (f>>x>>y)
    {d=cmmdc(x,y);
    g<<x/d<<' '<<y/d<<' \n' ; }
```

Varianta PASCAL

```
function cmmdc(a:word; b:word):word;

program p5;
var x,y,d:word;
    f,g:text;
function cmmdc(a:word; b:word):word;
begin
  assign (f,'numere.in');
  assign (g,'numere.out');
  reset(f);rewrite(g);
  while not eof(f) do
    begin
      readln(f,x,y);
      d:=cmmdc(x,y);
      writeln(g,x div d,' ',y div d);
```

```
f.close();g.close();}
```

```
end;
close(f);close(g);
end.
```

Varianta <86>

SUBIECTUL I

1. c

2.

a) 1 2 3 4 0 1 2

b) 21 (orice nr. mai mare ca 20)

c)

Varianta C/C++

```
#include<iostream.h>
void main()
{unsigned n,k,i;
 cout<<"n=";cin>>n;
 cout<<"k=";cin>>k;
 for(i=1;i<=n;i++)
  if (i/k==0) cout<<i<<' ';
  else cout<<i%k<<' ';
}
```

d)

citește n,k

$i \leftarrow 1$

repetă

[dacă $[i/k]=0$ atunci scrie i

[altfel scrie $i\%k$

$i \leftarrow i+1$

până când $i > n$

SUBIECTUL II

1. a

2. a

3.

0 1 1 1 0 0

1 0 0 0 0 0

1 0 0 0 1 1

1 0 0 0 0 0

0 0 1 0 0 0

0 0 1 0 0 0

4. Intensiv^{2 9} Neintensiv⁷

5. Vom completa simultan câte două linii, una de la stânga la dreapta, cealaltă învers.

Varianta C/C++

```
#include<iostream.h>
int n,m,a[11][11];
void main()
{int i,j,k=1;
 do{cout<<"n=";cin>>n;}while(n==0||n>10)
 m=n*2;

 for(i=1;i<=n;i++)
  {a[2*i-1][j]=k++;
   a[2*i][m+1-j]=k++;
```

Varianta PASCAL

```
program p1;
var n,k,i:word;
begin
 write('n=');read(n);
 write('k=');read(k);
 for i:=1 to n do
  if i div k=0 then write(i,' ')
  else write(i mod k,' ');
end.
```

Varianta PASCAL

```
program p2;
var n,m,i,j,k:word;
    a:array[1..10,1..10] of word;
begin
 write('n=');read(n);
 m:=n*2;

 for i:=1 to n do
  begin a[2*i-1][j]:=k;inc(k);
        a[2*i][m+1-j]:=k;inc(k);
```

```

    }
    for(i=1;i<=m;i++)
        { for(j=1;j<=m;j++)
            cout<<a[i][j]<<' ';
          cout<<endl; }
}

```

```

end;
for i:=1 to n do begin
    for j:=1 to m do
        write(a[i][j], ' ');
    writeln;
end;end.

```

SUBIECTUL III

1. a

2. ***##

3.

a)

Varianta C/C++

Intensiv

```

void numar(unsigned long
&n,unsigned c1,unsigned c2)
{unsigned long p=1;
 while(p<=n)
    {if(n/p%10==c1)
        n=(n/(p*10)*10+c2)*p+n%p;

        p*=10;}
}

```

Neintensiv

```

unsigned long numar(unsigned long
n,unsigned c)
{unsigned long p=1;
 while(p<=n)
    {if(n/p%10==c)
        n=(n/(p*10)*10+c/2)*p+n%p;
        p*=10;}
return n;}

```

4.

Varianta C/C++

Intensiv

```

#include<iostream.h>
#include<fstream.h>
#include<conio.h>
void main()
{
unsigned long x,max1,max2;
ifstream f("bac.txt");
max1=max2=-1000000000;
while(f>>x)
    if(x%2==0)
        if(max1<x){max2=max1;max1=x;}
        else if(max2<x)max2=x;
cout<<max1<<' '<<max2;
f.close();
}

```

Neintensiv

```

#include<iostream.h>
#include<fstream.h>

```

Varianta PASCAL

Intensiv

```

procedure numar(var
n:longint;c1,c2:word);
var p:longint;
begin p:=1;
 while p<=n do begin
    if(n div p) mod 10=c1 then
        n:=(n div (p*10)*10+c2)*p+n mod p;
        p:=p*10; end;
end;

```

Neintensiv

```

function numar(n:longint;
c:word):longint;
var p:longint;
begin p:=1;
 while p<=n do begin
    if(n div p) mod 10=c then
        n:=(n div(p*10)*10+c div 2)*p+n mod
p;
        p:=p*10; end;
numar:=n;
end;

```

Varianta PASCAL

Intensiv

```

program p3;
var x,max1,max2:longint;
f:text;
begin
assign (f,'bac.txt');reset(f);
max1:=-maxlongint;max2:=-maxlongint;
while not eof(f) do
    begin read(f,x);
        if x mod 2=0 then
            if max1<x then
                begin max2:=max1;max1:=x;end
            else if max2<x then max2:=x;
        end;
write(max1,' ',max2);

close(f);end.

```

Neintensiv

```

program p4;
var x,min1,min2:longint;

```

```
#include<conio.h>
void main()
{
unsigned long x,min1,min2;
ifstream f("bac.txt");
min1=min2=1000000000;
while(f>>x)
    if(min1>x){min2=min1;min1=x;}
    else if(min2>x)min2=x;
cout<<min1<<' '<<min2;
f.close();}
```

```
f:text;
begin
assign (f,'bac.txt');reset(f);
min1:=maxlongint;min2:=maxlongint;
while not eof(f) do
begin read(f,x);
if min1>x then
begin min2:=min1;min1:=x;end
else if min2>x then min2:=x;
end;
write(min1,' ',min2);
close(f);end.
```

b) Citim pe rând câte o valoare din fișier, verificăm dacă îndeplinește proprietatea cerută și în caz afirmativ, îl comparăm cu **max1/min1** respectiv cu **max2/min2**, înlocuindu-le când e cazul. Algoritmul este liniar și utilizează doar date simple.

Varianta <87>

SUBIECTUL I

1. d

2.

a) 2

b) 98, 91, 84

c)

Varianta C/C++

```
#include<iostream.h>
void main()
{unsigned a,b,c,x;
cout<<"a=";cin>>a;

cout<<"b=";cin>>b;
cout<<"c=";cin>>c;
while(a!=b && a!=c)
{x=a;
if (x>b) x=b;
if (x>c) x=c;
if (x!=a) a-=x;
if (x!=b) b-=x;
if (x!=c) c-=x;
}
cout<<a;}
```

d)

```

gitește a,b,c
dacă a ≠ b sau a ≠ c atunci
┌ repetă ← a
├ ┌ dacă x > b atunci x ← b
├ ┌ ┌ dacă x > c atunci x ← c
├ ┌ ┌ ┌ dacă x ≠ a atunci a ← a-x
├ ┌ ┌ ┌ ┌ dacă x ≠ b atunci b ← b-x
├ ┌ ┌ ┌ ┌ ┌ dacă x ≠ c atunci c ← c-x
├ ┌ ┌ ┌ ┌ ┌ ┌ până când a=b și a=c
├ ┌ ┌ ┌ ┌ ┌ ┌ ┌ scrie a
```

Varianta PASCAL

```
program p1;
var a,b,c,x:word;
begin
write('a=');read(a);
write('b=');read(b);
write('c=');read(c);
while(a<>b) and (a<>c)
begin x:=a;
if x>b then x:=b;
if x>c then x:=c;
if x<>a then a:=a-x;
if x<>b then b:=b-x;
if x<>c then c:=c-x;
end;
write(a);end.
```

SUBIECTUL II

1. a

2. b

3. 1 2

4. Intensiv2 -1 Neintensiv45

5. Parcurgem triunghiul de sub diagonala principala, determinand suma si numarul valorilor strict pozitive. Daca exista astfel de valori, afisam media lor, convertind suma la float pentru a calcula media cu zecimale.

Varianta C/C++

```
#include<iostream.h>
int n, a[21][21];
void main()
{int i, j, k=0, s=0;
do{cout<<"n="; cin>>n; }while (n==0 || n>20);
for(i=1; i<=n; i++)
for(j=1; j<=n; j++)
do{cout<<"a["<<i<<"] ["<<j<<"]=";
cin>>a[i][j];}
while (a[i][j]<=-100 || a[i][j]>100);
for(i=2; i<=n; i++)
for(j=1; j<i; j++)
if(a[i][j]>0)
{s+=a[i][j]; k++;}
if(k>0) cout<<float(s)/k;
else cout<<"sub diag. princ. nu exista
valori pozitive";
}
```

Varianta PASCAL

```
program p2;
var n, i, j, k, s: integer;
a: array[1..20, 1..20] of
integer;
begin
k:=0; s:=0;
write('n='); read(n);
for i:=1 to n do
for j:=1 to n do begin
write('a['<i, ', ', 'j, ']=');
read(a[i][j]); end;
for i:=2 to n do
for j:=1 to i-1 do
if a[i][j]>0 then
begin s:=s+a[i][j];
inc(k); end;
if k>0 then writeln(s/k)
else writeln('sub diag.
princ. nu exista valori
pozitive');
end.
```

SUBIECTUL III

- 1. b
- 2. 43211234
- 3.

a)

Varianta C/C++

Intensiv

```
unsigned reduce(unsigned long a,
unsigned long b)
{unsigned c, s=0;
while(a>0)
{c=a%10;
if(c>0 && b%c!=0) s+=c;
a/=10;
}
return s;
}
```

Neintensiv

```
unsigned long numar(unsigned long
a,
unsigned b)
{return a/pow(10,b) * pow(10,b);
}
```

Varianta PASCAL

Intensiv

```
function reduce(a: longint;
b: longint): word;
var c, s: word;
begin s:=0;
while a>0 do
begin c:=a mod 10;
if (c>0) and (b mod c<>0) then
s:=s+c;
a:=a div 10; end;
reduce:=s;
end;
```

Neintensiv

```
function numar(a: longint; b: word):
longint;
var p: longint;
begin p:=1;
while b>0 do
begin a:=a div 10; p:=p*10; dec(b);
end;
numar:=a*p; end;
```

4.

a) Varianta C/C++

Intensiv

```
#include<iostream.h>
#include<fstream.h>

int sume_egale(unsigned long n)
{int sp,si;
  sp=si=0;
  while(n>0)
    {if(n%2==0)sp+=n%10;
     else si+=n%10;
     n/=10;}
  return sp==si;
}

void main()
{
  unsigned long x,min, ap=0;
  ifstream f("bac.txt");
  min=1000000000;
  while(f>>x)
    if(sume_egale(x))
      if(x<min){min=x;ap=1;}
      else if(x==min)ap++;

  cout<<min<<' '<<ap;
  f.close();
}
```

Neintensiv

```
#include<iostream.h>
#include<fstream.h>

void main()
{unsigned x,k,max,ap=0;
  ifstream f("bac.txt");
  max=-1000000000;
  while(f>>x)
    if(x<=k)
      if(x>max){max=x;ap=1;}
      else if(x==max)ap++;

  if(max>0) cout<<max<<' '<<ap;
  else cout<<0;
  f.close();}
```

Varianta PASCAL

Intensiv

```
program p3;
var x,min,ap:longint;

f:text;
function sume_egale(n:longint):boolean;
var sp,si:word;
begin sp:=0;si:=0;
  while n>0 do
    begin if n mod 2=0
      then sp:=sp+n mod 10
      else si:=si+ n mod 10;
      n:=n div 10; end;
  sume_egale:=sp=si; end;
begin ap:=0;
  assign (f,'bac.txt'); reset(f);
  min:=maxlongint;
  while eof(f) do begin
    read(f,x);
    if sume_egale(x) then
      if x<min then
        begin min:=x;ap:=1; end
        else if x=min then inc(ap);
    end;
  write(min,' ',ap);
  close(f); end.
```

Neintensiv

```
Var x,k,max,ap:longint;
f:text;

begin ap:=0;
  assign (f,'bac.txt'); reset(f);
  max:=-maxlongint;
  while eof(f) do begin
    read(f,x);
    if x<=k then
      if x>max
        then begin max:=x; ap:=1; end
        else if x=max then inc(ap);
    end;
  if max>0 then write(max,' ',ap)
  else write(0);
  close(f); end.
```

b) Citim pe rând câte o valoare din fișier, verificăm dacă îndeplinește proprietatea cerută și în caz afirmativ, o comparăm cu minimumul/maximumul, reținându-l dacă este cazul respectiv, crescând nr. de apariții în caz de egalitate. Algoritmul este liniar și utilizează doar date simple.

Varianta <88>

SUBIECTUL I

1. b

2.

a) 246531

b) 13579 (orice nr. cu 5 cifre impare)

c)

Varianta C/C++

```
#include<iostream.h>
void main()
{unsigned long a,b,p; unsigned c;

cout<<"a=";cin>>a;
while (a!=0)
{c=a%10;
if (a%2==0) b=b+c*p;
else b=b*10+c;
a/=10;p*=10;}
cout<<b;}
```

Varianta PASCAL

```
program p1;
type natural=0..maxlongint;
var a,b,p:natural;

begin
write('a=');read(a);
p:=1; b:=0;
while a<>0 do begin
c:=a mod 10;
if a mod 2=0 then b:=b+c*p
else b:=b*10+c;
a:=a div 10; p:=p*10;
end;
write(b); end.
```

d)

```
citeste a; b ← 0
repetăc ← a%10
[ dacăa%2=0 atunci b ← b+c*p
altfel b ← b*10+c
a ← [a/10]; p ← p*10
pânăcând a=0
scrie b
```

SUBIECTUL II

1. a

3. b 4 5 6

4. Intensiv2 1 Neintensiv24

5. Parcurgem triunghiul de deasupra diagonalei principale, determinând suma și numărul valorilor strict pozitive. Dacă există astfel de valori, afișăm media lor, convertind suma la float pentru a calcula media cu zecimale.

Varianta C/C++

```
int n, a[21][21];
void main()
{int i,j,k=0,s=0;

do{cout<<"n=";cin>>n;}while (n==0||n>20);
for (j=1;j<=n;j++)
do{cout<<"a["<<i<<"] ["<<j<<"]=";
cin>>a[i][j];}
while (a[i][j]<-100||a[i][j]>100);
for (i=1;i<=n;i++)
for (j=i+1;j<=n;j++)
if (a[i][j]>0)
{s+=a[i][j]; k++;}
if (k>0) cout<<float(s)/k;
else cout<<"sub diag. princ. nu
exista valori pozitive";
}
```

Varianta PASCAL

```
program p2;
var n,i,j,k,s:integer;
a:array[1..20,1..20] of
integer;

begin
s:=0;
write('n=');read(n);
for i:=1 to n do
for j:=1 to n do begin
write('a['<i','<j,']=');
read(a[i][j]);end;
for i:=1 to n-1 do
for j:=i+1 to n do
if a[i][j]>0 then
begin s:=s+a[i][j];
inc(k);end;
if k>0 then write(s/k)

else write('sub diag.
princ. nu exista valori
pozitive');
end.
```

SUBIECTUL III

1. a

2. -2

3. Pornind de la ultima cifră spre prima, se elimină prin operații aritmetice fiecare cifră care aparține intervalului.

Varianța C/C++

Intensiv

```
void numar(unsigned long n, unsigned
c1, unsigned c2, unsigned long &x)
{unsigned long p=1;
 unsigned c;
 x=n;
 while (p<=x)
 {c=x/p%10;
 if (c>=c1&& c<=c2)
 x=x/(p*10)*p+x%p;
 }
 else p*=10;}
```

Neintensiv

```
void numar(unsigned long n, unsigned
c1, unsigned c2, unsigned &x)
{unsigned c; long p=1;
 x=0;
 while (x>0)
 {c=x/p%10;
 if (c>=c1&& c<=c2) x+=c;
 p*=10;}
 }
```

Varianta PASCAL

Intensiv

```
procedure
 numar(n:longint; c1, c2:byte;
 var x:longint);
var p:longint; c:byte;
begin p:=1;
 x:=n;
 while p<=x do
 begin c:=x div p mod 10;
 if (c>=c1) and (c<=c2)
 then x:=x div (p*10)*p+x mod p
 end;
```

Neintensiv

```
procedure
 numar(n:longint; c1, c2:byte;
 var x:byte);
var p:longint; c:byte;
begin x:=0;
 while x>0 do begin
 c:=x div p mod 10;
 if (c>=c1) and (c<=c2) then
 x:=x+c;
 p:=p*10; end;
 end;
```

4.

a) Varianta C/C++

Intensiv

```
#include<iostream.h>
#include<fstream.h>
#include<math.h>
int prim(int n)
{int d;
 for(d=3; d<=sqrt(n); d+=2)
 if(n%d==0) return 0;
 }
void main()
{
 int x, c[100];
 ifstream f("bac.txt");
 for(x=0; x<100; x++) c[x]=0;
 while(f>>x)
 c[x]++;
 x=99;
 while(x>2&&(!c[x]||!prim(x))) x-=2;
 if(x>2) cout<<x<<' '<<c[x];
 else if(c[2]!=0) cout<<2<<' '<<c[2];
 else cout<<"fisierul nu contine
 numere prime";
 f.close();}
```

Varianta PASCAL

Intensiv

```
program p3;
var x:integer;
 c:array[1..100] of integer;
 f:text;
function prim(n:integer):Boolean;
var d:integer;
begin
 prim:=true;
 while (d<=sqrt(n)) and (n mod d<>0)
 do
 d:=d+2;
 prim:=d>sqrt(n);
 end;
begin
 assign (f, 'bac.txt'); reset(f);
 while not eof(f) do begin
 read(f, x);
 inc(c[x]); end;
 x:=99;
 while (x>2) and ((c[x]=0) or not
 prim(x)) do dec(x, 2);
 if x>2 then write(x, ' ', c[x])
 else if c[2]<>0 then write(2, '');
```



```
,c[2]) else write('fis.nu contine
nr.prime');
close(f);end.
```

b) Citim pe rând câte o valoare și creștem frecvența acesteia cu 1, construind astfel vectorul $c[100]$. Parcurgem descrescător din 2 în doi valorile de la 99 determinând cel mai mare număr prim care a apărut în fișier (are frecvența mai mare ca 0). Algoritmul este eficient din punct de vedere al execuției – determină liniar frecvența, verifică proprietatea de număr prim doar pentru valori impare, și gestionează eficient memoria deoarece nu reținem toate valorile din fișier, ci doar frecvența numerelor de cel mult două cifre.

Neintensiv

```
#include<iostream.h>
#include<fstream.h>
#include<math.h>
int prim(int n)
{int d;
 for(d=3;d<=sqrt(n);d+=2)
  if(n%d==0) return 0;
 return 1;
}
void main()
{
int x,max=0,a;
ifstream f("bac.txt");
while(f>>x)
  if(x>max)max=x;
a=max;
if(a%2==0)a++;
while(!prim(a))a+=2;
cout<<a;
f.close();}
```

Neintensiv

```
program p4;
var x,max,a:integer;
    f:text;
function prim(n:integer):Boolean;
var d:integer;
begin prim:=true;
    d:=3;
    while (d<=sqrt(n))and(n mod d<>0)
do
    d:=d+2;
    prim:=d>sqrt(n);
end;
begin max:=0;
  assign (f,'bac.txt');reset(f);
  while not eof(f) do begin
    read(f,x);
    if x>max then max:=x;
  end;
  a:=max;
  if a mod 2=0 then inc(a);
  while not prim(x) do inc(a,2);
  write(a);
  close(f);end.
```

Citim pe rând câte o valoare din fișier și determinăm maximul. Căutăm apoi o valoare primă impară mai mare decât max. Algoritmul este eficient din punct de vedere al execuției – determină liniar maximul, verifică proprietatea de număr prim doar pentru valori impare, și gestionează eficient memoria deoarece nu reținem toate valorile din fișier.

Varianta <89>**SUBIECTUL I**

1. a

2.

a) 1012141

b) 12468

c)

Varianta C/C++

```
#include<iostream.h>
void main()
{unsigned long n,t,r;
 cout<<"n=";cin>>n;
 t=n; r=0;
 while (t>0)
  {if ((t%10)%2==1) r=r*10+1;
```

Varianta PASCAL

```
program p1;
type natural=0..maxlongint;
var n,t,r:natural;
begin
  write('n=');read(n);
  t:=n; r:=0;
  while t>0 do begin
```

```

        else r=r*10+t%10;
        t/=10;}
n=0;
while(r>0)
    {n=n*10+r%10;r/=10;}
cout<<n;}

```

```

        if t mod 10 mod 2=0 then
r:=r*10+1
        else r:=r*10+t mod 10;
        t:=t div 10;
end;
while t<0 do begin
    n:=n*10+r mod 10;
    r:=r div 10;
end;
write(n); end.

```

d)

citește n

$t \leftarrow n; r \leftarrow 0$

```

repetă
[
    dacă(t%10) %2=1 atunci r ← r*10+1
    altfel r ← r*10+t%10
    t ← [t/10]
pânăcând t=0
n ← 0
repetă n ← n*10+r%10
    r ← [r/10]
pânăcând r=0

```

SUBIECTUL II

1. Intensiv a Neintensivd

2. a

3. 0 1 1 0 0 0 0
 1 0 0 1 0 0 1
 1 0 0 0 1 1 0
 0 1 0 0 0 0 0
 0 0 1 0 0 0 0
 0 0 1 0 0 0 0
 0 0 1 0 0 0 0
 0 1 0 0 0 0 0

4. Intensiv 2 23 Neintensiv28

5. Utilizând variabilele simple x,y,z, se calculeazătermenii Fibonacci și se rețin în matrice de la stânga la dreapta pe liniile de ordin impar, respective de la dreapta la stânga în liniile de ordin par.

Varianta C/C++

```

unsigned n,m;
unsigned long a[7][7],x,y,z;
int main()
{int i,j;
do{cout<<"n=";cin>>n;}while(n==0||n>6);
x=0;y=1;
for(i=1;i<=n;i++)
    {if(i%2==1)
        for(j=1;j<=n;j++)
            {a[i][j]=y;z=x+y;x=y;y=z;}
        else
            for(j=n;j>=1;j--)
                {a[i][j]=y;z=x+y;x=y;y=z;}
    }
for(i=1;i<=n;i++)

```

Varianta PASCAL

```

program p2;
var n,i,j,x,y,z:word;
    a:array[1..7,1..7] of longint;
begin
    write('n=');read(n);
    x:=0;y:=1;
    for i:=1 to n do
        if i mod 2 =0 then
            for j:=1 to n do begin
                a[i][j]:=y;z:=x+y;x:=y;y:=z; end
            else for j:=n downto 1 do begin
                a[i][j]:=y;z:=x+y;x:=y;y:=z; end;
        for i:=1 to n do begin
            for j:=1 to n do
                write(a[i,j], ' ');

```

```

    {for (j=1; j<=n; j++)
        cout<<a[i][j]<<' ';
        cout<<endl; }
}

```

SUBIECTUL III**1: 90****3.**

a) Pornind de la ultima cifră spre prima, se înlocuiește cu c2 orice cifră egală cu c1, iar cifrele de după aceasta cu 0.

Varianta C/C++

```

void numar(unsigned long &n, unsigned
c1, unsigned c2)
{unsigned long p=1;
while (p<=n)
{if (n/p%10==c1)
n=(n/(p*10)*10+c2)*p;
p*=10;}
}

```

4.**Intensiv****a) Varianta C/C++**

```

#include<iostream>
#include<fstream>
#include<conio.h>
void main()
{ int k,x,c[100],i;
ifstream f("date.txt");
for (x=0;x<100;x++) c[x]=0;
cout<<"k=";cin>>k;
while (f>>x)
c[x]++;
i=0;
for (x=99;x>k;x--)
if (c[x]>0)
{cout<<x<<' '; i++;
if (i%20==0) cout<<endl;
}
f.close();
}

```

Varianta PASCAL

```

procedure numar(var n:longint;
c1,c2:byte);
var p:longint;
begin n:=0;
while n>0 do begin
if n div p mod 10=c1 then
n:=(n div (p*10)*10+c2)*p;
p:=p*10;end;
end;

```

Intensiv**Varianta PASCAL**

```

program p3;
var x,k,i:integer;
c:array[1..100] of integer;
f:text;
begin
assign (f,'bac.txt');reset(f);
write('k=');read(k);
while not eof(f) do begin
read(f,x);
inc(c[x]);end;
i:=0;
for x:=99 downto k+1 do
if c[x]>0 then begin
write(x,' ');inc(i);
if i mod 20=0 then writeln;
end;
close(f);
end.

```

b) Citim pe rând câte o valoare și creștem frecvența acesteia cu 1, construind astfel vectorul c[100]. Pentru afișarea valorilor cerute, parcurgem descrescător valorile de la 99 la k+1 și verificăm dacă au apărut în fișier. Algoritmul este liniar, și gestionează eficient memoria deoarece nu reținem toate valorile din fișier, ci doar frecvența numerelor de cel mult două cifre.

Neintensiv**a) Varianta C/C++**

```

#include<iostream>
#include<fstream>
#include<conio.h>
unsigned nrdiv(unsigned n)
{unsigned d=2,nd=1,c;
while (n>1)
{c=0;
while (n%d==0) {c++;n/=d;}
}
}

```

Varianta PASCAL**Neintensiv**

```

program p4;
var x,nd,k,i:integer;
c:array[1..100] of integer;
f:text;
function nrdiv(n:word):word;
var d,nd,c:word;
begin
nd:=1;d:=2;

```

```

    nd*=(c+1);
    d++;}
return nd;
}
void main()
{unsigned x,c[100],nd;
 ifstream f("bac.txt");
 for(x=1;x<100;x++)c[x]=0;
 cout<<"k=";cin>>k;
 while(f>>x)
    if(c[x]==0){nd=nrdiv(x);
    if(nd>k)c[x]++;}
 for(x=10;x<100;x++)
    if(c[x]>0)cout<<x<<' ';
 f.close();}

```

```

while n>1 do
    begin c:=0;
while n mod d=0 do
    begin inc(c);n:=n div d;end;
nd:=nd*(c+1); inc(d); end;
nrdiv:=nd; end;
begin
assign (f,'bac.txt');reset(f);
write('k=');read(k);
while not eof(f) do begin
    read(f,x);
    if c[x]=0 then begin
nd:=nrdiv(x);
    if nd>=k then inc(c[x]) end;
end;
for x:=10 to 99 do
    if c[x]>0 then write(x,' ');
close(f);
end.

```

b) Se citesc pe rând valorile din fișier, se determină valorile cu cel puțin k divizori – utilizând descompunerea în factori primi - și se rețin în vectorul caracteristic. Programul utilizează doar date simple și determină eficient numărul de divizori și aparițiile valorilor în fișier.

Varianta <90>

SUBIECTUL I

1. c

2.

a) 107 117

b) a=230, b=237, k=8; a=138, b=100, k=6

c)

citește a,b,k

```

t ← a
p ← 0
[
    repetă
    [
        dacă k=t%10 atunci
            scrie t
            p ← 1
            ←
        până când t>b
        dacă p=0 atunci
            scrie -1
    ]
]

```

d)

Varianta C/C++

```

#include<iostream.h>
void main()
{unsigned a, b, k, t;
 int p;
 cout<<"a=";cin>>a;
 cout<<"b=";cin>>b;
 cout<<"k=";cin>>k;
 t=a;
 p=0;

```

Varianta PASCAL

```

program pl;
var a,b,k,t:word;
    p:integer;
begin
    write('a=');read(a);
    write('b=');read(b);
    write('k=');read(k);
    p:=0; t:=a;
    while t<=b do begin

```

```

while (t<=b)
    {if      (k==t%10){cout<<t<<'
';p=1;}
    t=t+1;}
if (p==0) cout<<-1;}

```

```

if k=t mod 10 then
    begin write(t, ' ');p:=1;end;
    t:=t+1;end;
if p=0 then write(-1);
end.

```

SUBIECTUL II

1. a

2. b

3. 1 2

4. Intensiv 2 16 Neintensiv 10

5. Vezi variantele 87 și 88, problemele II 5.

Varianta C/C++

```

#include<iostream.h>
int n, a[21][21];
void main()
{int i, j, k=0;
 float m1=0, m2=0;

do{cout<<"n="; cin>>n;}while (n==0 || n>20);
for (i=1; i<=n; i++)
    for (j=1; j<=n; j++)
do{cout<<"a["<<i<<"]["<<j<<"]="";
cin>>a[i][j];}
while (a[i][j]<-100 || a[i][j]>100);
for (i=1; i<=n; i++)
    for (j=i+1; j<=n; j++)
        if (a[i][j]>0)
            {m1+=a[i][j]; k++;}
if (k>0) {m1/=k; k=0;
for (i=2; i<=n; i++)
    for (j=1; j<i; j++)
        if (a[i][j]>0)
            {m2+=a[i][j]; k++;}
if (k>0) {m2/=k; cout<<m1-
m2;}
else cout<<"nu se poate
calcula m2";}
else cout<<"nu se poate calcula
m2";
}

```

Varianta PASCAL

```

program p2;
var n, i, j, k:word;
    m1, m2:real;
    a:array[1..20, 1..20] of word;
begin
    k:=0; m1:=0; m2:=0;
    write('n='); read(n);
    for i:=1 to n do
        for j:=1 to n do begin
            write('a['<i, ', ', j, ']=');
            read(a[i][j]); end;
        for i:=1 to n-1 do
            for j:=i+1 to n do
                if a[i][j]>0 then
                    begin
                        m1:=m1+a[i][j]; inc(k); end;
                    if k>0 then begin m1:=m1/k; k:=0;
                        for i:=2 to n do
                            for j:=1 to i-1 do
                                if a[i][j]>0 then
                                    begin
                                        m2:=m2+a[i][j]; inc(k); end;
                                        if k>0 then begin
                                            m2:=m2/k; write(m1-2); end
                                        end
                                    else write('nu se poate det. M2')
                                        end
                                    else write('nu se poate det. m1');
                                end.

```

SUBIECTUL III

1. a

2. dcba

3. Este posibilă înlocuirea proprietății crescător/descrescător cu strict crescător/descrescător.

Varianta C/C++

Intensiv

```

int f(int n, int a[201])
{int i;
 if (a[1]==a[2])//verif. propr. de sir
 constant
    {i=3; while (i<=n && (a[i-1]==a[i])) i++;

```

Varianta PASCAL

Intensiv

```

type vector=array[1..200] of
integer;
function
f(n:integer; a:vector):integer;
var i:integer;
begin

```

```

        if(i>n) return 0;}
if(a[1]<=a[2]) //pt. sir crescator
    {i=3;while(i<=n&&(a[i-1]
<=a[i])i++;
        if(i>n) return 1;
        else return -1;}
//verif. propr. de sir descrescator
i=3;while(i<=n&&(a[i-1]<=a[i])i++;
if(i>n) return 2;
    else return -1;
}

```

Neintensiv

```

int f(int n, int a[201])
{int i,max,min;

    max=min=a[1];i++;
    if(a[i]>max)max=a[i];
    else if(a[i]<min)min=a[i];
    if(max-min>1000)return 1;
    else return 0;
}

```

```

if a[1]=a[2] then begin i:=3;
while(i<=n)and(a[i-1]=a[i])      do
inc(i);
if i>n then f:= 0;end;
if a[1]<=a[2] then begin i:=3;
while(i<=n)and(a[i-1]<=a[i])do
inc(i);
if i>n then f:=1
else f:=-1
end;
i:=3;while(i<=n)and(a[i-1]
<=a[i])do inc(i);
if i>n then f:=2
else f:=-1;
end;

```

Neintensiv

```

type    vector=array[1..200]    of
integer;

function
    f(n:integer;a:vector):integer;
var i,max,min:integer;
begin max:=a[1]; min:=a[1];
for i:=2 to n do
    if a[i]>max then max:=a[i]
    else if a[i]<min then min:=a[i];
if max-min>1000 then f:=1
else f:=0;
end;

```

4. Propunem eliminarea cuvintelor “cele mai mari” din textul problemei.

a) Varianta C/C++

Intensiv

```

#include<iostream.h>
#include<fstream.h>
#include<math.h>
#include<conio.h>
void main()
{int x,y,smax=0,xmax,ymax;
ifstream f("bac.txt");
f>>x;
smax=0;

while(x>0&&y>0)
    if(x+y>smax){smax=x+y;xmax=x;ymax=y;}
    else
if(x+y==smax&&abs(x-y)>abs(xmax-ymax))
    {xmax=x;ymax=y;}

    x=y;
}
if(xmax>ymax)
    cout<<xmax<<' '<<ymax;
else cout<<ymax<<' '<<xmax;
f.close();
}

```

Neintensiv

```

#include<iostream.h>
#include<fstream.h>

```

Varianta PASCAL

Intensiv

```

program p3;
var x,y,smax,xmax,ymax:integer;
    c:array[1..100] of integer;
    f:text;
begin
assign (f,'bac.txt');reset(f);
read(f,x);smax:=0;
while not eof(f) do begin
    read(f,y);

    if(x>0)and(y>0) then
    if x+y>smax then begin
smax:=x+y;xmax:=x;ymax:=y;end
    else
if(x+y=smax)and(abs(x-y)>abs(xmax-
ymax))
    then begin xmax:=x;ymax:=y;
end;
    x:=y;
end;
if xmax>ymax then
    write(xmax,' ',ymax)
    else write(ymax,' ',xmax);

close(f);end.
Neintensiv
program p4;

```

```
#include<math.h>
unsigned nrdiv(unsigned n)
{unsigned d=2,c,nd=1;
 while(n>1)
 {c=0;
  while(n%d==0){c++;n/=d;}
  nd*=(c+1);
  d++;}
 return nd;
}
void main()
{unsigned x,max,s,nd;
 ifstream f("bac.txt");
 max=0;
 while(f>>x)
 {nd=nrdiv(x);
  if(nd>max){max=nd;s=x;}
  else if(nd==max&& x<s) s=x;}

 f.close();
}
```

```
var x,nd,max,s:integer;
f:text;
function nrdiv(n:word):word;
var d,nd,c:word;
begin
 nd:=1;d:=2;
 while n>1 do
 begin c:=0;
  while n mod d=0 do
 begin inc(c);n:=n div d;end;
 nd:=nd*(c+1); inc(d); end;
 nrdiv:=nd; end;
begin
 assign (f,'bac.txt');reset(f);
 while not eof(f) do begin
 read(f,x);nd:=nrdiv(x);
 if nd>max then begin
 max:=nd;s:=x;
 endelse if (nd=max)and(x<s) then
 s:=x;
 end;
 write(x,' ');
 close(f);
end.
```

b) Intensiv Se citesc pe rând valorile din fișier, reținând în fiecare moment două valori x, y , a căror sumă se compară cu s_{max} atunci când sunt pozitive, modificându-se valoarea s_{max} și reținând cele două valori pentru care s-a obținut o sumă mai mare, iar în caz de egalitate se verifică dacă modulul diferenței este mai mare decât cel precedent, actualizându-se x_{max} și y_{max} . Se obține un algoritm liniar ce prelucrează doar variabile simple, fiind eficient ca timp de execuție și spațiu de memorie utilizat.

Neintensiv Se citesc pe rând valorile din fișier, se determină valorile cu număr maxim de divizori – utilizând descompunerea în factori primi - iar din acestea cea mai mică. Programul utilizează doar date simple și determină eficient numărul de divizori.

Varianta <91>

SUBIECTUL I

1. d

b) citește z

$$x \leftarrow |z|$$

$$y \leftarrow 0$$

cât timp $x \neq y$ execută

$$y \leftarrow x$$

$$x \leftarrow [(x+z/x)/2]$$

scrie x

c) Varianta C/C++

```
#include<iostream.h>
#include<math.h>
void main()
{int x, y, z;
 cout<<"z=";cin>>z;
 z=abs(z);
 x=1;
```

Varianta PASCAL

```
program p1;
var x,y,z:integer;
begin
 write('z=');read(z);
 z:=abs(z);
 x:=1;
 repeat
```

```
do{y=x;
    x=(x+z/x)/2;
}
while(x!=y);
cout<<"x="<<x;
}
```

```
y:=x; x:=(x+z div x)div 2;
until x=y;
write('x=',x);
end.
```

d) 1

SUBIECTUL II

1. Intensivd Neintensivb

2. a

3. Intensiv

Varianta C/C++

```
for (j=1; j<=10; j++)
    A[j][3]=100;
```

Neintensiv 3

4. Intensiv

da
dacarba

Neintensiv iarba

5. Se creează lista dublu înl ănuțită inserând câte un caracter la sfâr șitul listei, apoi se afișeazăinforma țile de la ultimul nod spre primul.

Varianta C/C++

Intensiv

```
#include<iostream.h>
#include<conio.h>
#include<stdlib.h>
#include<string.h>
struct nod{char info;
            nod*prec,*urm;}*p, *u;
void creare(char s[31],nod*&p,nod*&u)
{nod *q;
 u=p=new nod;
 p->info=s[0];p->urm=p->prec=NULL;
 for (int i=1;i<strlen(c);i++)
     {q=new nod;
      q->info=s[i];
      q->urm=NULL;

      q->prec=u;
      u=q;}
}
void listare(nod*q)
{while(q)
 {cout<<q->info; q=q->prec;}
 cout<<endl;
}
void main()
{char s[31];
 cout<<"dati cuvantul ";cin>>s;
 creare(s,p,u);
 cout<<"lista este ";
 listare(u);
}
```

Varianta PASCAL

```
for j:=1 to 10 do
    a[j][3]:=100;
```

Varianta PASCAL

Intensiv

```
program p2;
type lista=^nod;
      nod=record info:char;
                prec,urm:lista;end;
      cuvant=string[30];
var p,u:lista;
    s:cuvant;
procedure creare(s:cuvant;var
p,u:lista);
var q:lista;i:byte;
begin new(p);u:=p;
      p^.info:=s[1];p^.urm:=nil;
      p^.prec:=nil;
      for i:=2 to length(s) do
          begin new(q);q^.info:=s[i];
                q^.urm:=nil;q^.prec:=u;
                u^.urm:=q;u:=q;end;
      end;
procedure listare(u:lista);
begin while u<>nil do begin
      write(u^.info); u:=u^.prec;
      end;
      writeln;
      end;
begin
write('Introduceti un cuvant:
');read(s);
creare(s,p,u);
write('lista este ');listare(u);
end.
```


Neintensiv

```
int n, a[21][21];
void main()
{int i, j;
do {cout<<"n=";cin>>n;}
while (n==0||n>10);
for(i=1;i<=n;i++)
for(j=1;j<=n;j++)
{cout<<"a["<<i<<"["<<j<<"]=";
cin>>a[i][j];a[0][j]+=a[i][j];}
j=1;
for(i=2;i<=n;i++)
if(a[i][0]>a[j][0])j=i;
cout<<j;
}
```

Neintensiv

```
program p3;
var n, i, j:integer;
a:array[0..20,0..20] of word;
begin
write('n=');read(n);
for i:=1 to n do
for j:=1 to n do begin
write('a['i',' ',j,']=');
read(a[i][j]);
a[0,j]:=a[0,j]+a[i,j]; end;
j:=1;
for i:=2 to n do
if a[i,0]>a[j,0] then j:=i;
write(j);
end.
```

SUBIECTUL III

1. d

2. **Intensiv** $i > 0$

Neintensiv 20105211251020

3.

a)

Varianta C/C++

```
int par(unsigned n);
```

b) Citim n, apoi pe rând, cele n numere; pentru fiecare valoare citită apelăm funcția par(x/10000) pentru a verifica dacă cel puțin una din primele patru cifre este pară și, în caz afirmativ, incrementăm un contor ce va fi afișat la final.

Varianta C/C++

```
#include<iostream.h>
int par(unsigned n);
void main()
{long unsigned n,i,x,c=0;
cout<<"n=";cin>>n;
for(i=1;i<=n;i++)
{ cout<<"x=";cin>>x;
if(par(x/10000)==1)c++;}
cout<<c;
}
```

Varianta PASCAL

```
function par(n:word):byte;
```

Varianta PASCAL

```
program p4;
type natural=0..maxlongint;
function par(n:word):byte;
{...}
var n,i,x,c:natural;
begin write('n=');read(n);c:=0;
for i:=1 to n do
begin write('x=');read(x);
if(par(x div 10000)=1)
then inc(c);end;
write(c);
end.
```

4) Citim pe rând câte o valoare din fișier, o comparăm cu valoarea maximă curentă; dacă este mai mare decât aceasta o reținem și inițializăm prima, respectiv ultima linie în care apare valoarea maximă cu nr. liniei curente. La întâlnirea unei valori egale cu cea maximă modificăm doar ultima linie în care apare maximul. Alg. este liniar - eficient d.p.d.v al timpului de execuție și lucrează doar cu variabile simple – eficient d.p.d.v al spațiului de memorie utilizat.

b) Varianta C/C++

```
#include<iostream.h>
#include<fstream.h>
int main()
{
unsigned i,primalinie,ultimalinie;
float x,max;
ifstream f("numere.txt");
```

Varianta PASCAL

```
program p5;
var i,primalinie,ultimalinie:word;
x,max:real;
f:text;
begin
assign (f,'numere.txt');reset(f);
max:=x;primalinie:=1;ultimalinie:=1;
```

```
f>>x;
max=x;
primalinie=ultimalinie=i=1;
while (f>>x)
{ i++;
  if (x>max) {max=x;

primalinie=ultimalinie=i;}
  else if (x==max)ultimalinie=i;
}
cout<<max<<' '<<primalinie<<'
'<<ultimalinie;
f.close();
getch();
return 0;}
```

```
i:=1;
while not eof(f) do begin
  read(f,x);inc(i);
  if x>max then begin max:=x;
primalinie:=i; ultimalinie:=i;
  end
  else if (x=max) then
ultimalinie:=i;
end;
write(max,' ',primalinie,' ',
ultimalinie);
close(f);
end.
```

Varianta <92>

SUBIECTUL I

1. a)

2.

a) algoritmul descrie adunarea a n numere naturale din intervalul închis [1,10]. Se va afișa 45/5 adică 9 precum și numărul de numere introduse în total (inclusive cele care nu se află în intervalul precizat mai sus), adică 9.

b) Ultima valoare afișată (nr) reprezintă numărul de numere introduse. Deci dacă numerele introduse sunt toate în [1,10] atunci nr va fi egal cu n. Deci pentru șirul n=3 și valorile 5,6,7 de exemplu, se va afișa 3.

c)

Varianta C/C++

```
#include<iostream.h>
int n;
float x;
void main()
{ cin>>n;
  int nr=0;
  int y=0;
  for(int i=1;i<=n;i++)
  { do
    {cin>>x;
     nr++;
    }while(x<1 || x>10);
    y=y+x;
  }
  cout<<y/n<<endl<<nr<<endl;
}
```

Varianta Pascal

```
var n,nr,x,y,i:integer;
begin
  readln(n);
  for i:=1 to n do
  begin
    repeat
      readln(x);
      nr:=nr+1;
    until ((x>=1) and (x<=10));
    y:=y+x;
  end;
  writeln(y div n);
  writeln(nr);
end.
```

d)

citește n (n număr natural nenul)

nr ← 0

y ← 0

pentru i ← 1, n execută

citește x (număr real)

nr ← nr + 1

cât timp (x < 1 sau x > 10)

citește x (număr real)

nr←nr+1

y←y+x

scrie [y/n]

scrie nr

SUBIECTUL II

1. **b)**
2. **b)**
3. **3 6 0 1 8 1 1 3 (intensiv)**
3. **3 6 0 1 3 1 (neintensiv)**
- 4.

Varianta C/C++

v.origine.x==0 &&v.origine.y==0 &&
v.extremitate.x==0&&v.extremitate.y==0

5. Explicații

Varianta C/C++

```
#include<iostream.h>
int n,m,a[51][51],i,j;
void afisare()
{for(int i=1;i<=m;i++)
  {for(int j=1;j<=n;j++)
    cout<<a[i][j]<<" ";
    cout<<endl;
  }
}
void muta_linia(int i)
{ for (int j=1;j<=n;j++)
  a[i-1][j]=a[i][j];
}
void muta_coloana(int j)
{ for (int i=1;i<=m;i++)
  a[i][j-1]=a[i][j];
}
void main()
{ cin>>m;
  cin>>n;
  for(i=1;i<=m;i++)
    for(j=1;j<=n;j++)
      cin>>a[i][j];
  int min=a[1][1],l,c;
  l=1;
  c=1;
  for(i=1;i<=m;i++)
    for(j=1;j<=n;j++)
      if(min>a[i][j])
        { min=a[i][j];
          l=i;
          c=j;
        }
  //eliminam linia l si coloana c din
  matrice
```

Varianta PASCAL

v.origine.x=0 and v.origine.y=0 and
v.extremitate.x=0 and
v.extremitate.y=0

Varianta PASCAL

```
var n,m,i,j,min,l,c:integer;
    a:array[1..51,1..51] of
integer;
procedure afisare;
var i:integer;
begin
  for i:=1 to m do
    begin
      for j:=1 to n do
        write(a[i][j],' ');
      writeln;
    end;
end;
procedure muta_linia(i:integer);
var j:integer;
begin
  for j:=1 to n do
    a[i-1][j]:=a[i][j];
end;
procedure muta_coloana(j:integer);
var i:integer;
begin
  for i:=1 to m do
    a[i][j-1]:=a[i][j];
end;
begin
  readln(m);
  readln(n);
  for i:=1 to m do
    for j:=1 to n do
      readln(a[i][j]);
  min:=a[1][1];
  l:=1;
  c:=1;
  for i:=1 to m do
```

```

for(i=l+1;i<=m;i++)
    muta_linia(i);
m--;
for(j=c+1;j<=n;j++)
    muta_coloana(j);
n--;
afisare();
}

```

```

for j:=1 to n do
    if(min>a[i][j]) then
        begin
            min:=a[i][j];
            l:=i;
            c:=j;
        end;
    {eliminam linia l si coloana c
din matrice}
    for i:=l+1 to m do
        muta_linia(i);
    m:=m-1;

    for j:=c+1 to n do
        muta_coloana(j);
    n:=n-1;
    afisare;
end.

```

SUBIECTUL III

1. a)

2.

a.) 01214

b.) 27 sau 28

3.

a)

Varianta C/C++

void ordonare(float x[101],int n)

b)

Varianta C/C++

```

#include<fstream.h>
float a[101],m,n;
void ordonare(float x[101],int
n)
{ for(int i=1;i<=n-1;i++)
    {float min=x[i];
    int poz=i;
    for(int j=i+1;j<=n;j++)
        if(min>x[j])
            { min=x[j];
            poz=j;
            }
        a[poz]=a[i];
        a[i]=min;
    }
}
void citire()
{
do
    { cin>>n>>m;
    }
while(n<1 ||n>100 || m<1 ||
m>n);

for (int i=1;i<=n;i++)

```

Varianta PASCAL

```

procedure ordonare(var
x:vector;n:integer);

```

Varianta PASCAL

```

type vector=array[1..101] of real;
var m,n,i:integer;
a:vector;
procedure ordonare(var
x:vector;n:integer);
var i:integer;
min:real;
poz,j:integer;
begin
for i:=1 to n-1 do
begin
min:=x[i];
poz:=i;
for j:=i+1 to n do
if(min>x[j]) then
begin
min:=x[j];
poz:=j;
end;
a[poz]:=a[i];
a[i]:=min;
end;end;
procedure citire;
var i:integer;

```

```

        cin>>a[i];
    }
void main()
{ citire();
  ordonare(a,n);
  for(int i=n-m+1;i<=n;i++)
      cout<<a[i]<<" ";
  cout<<endl;
  for(i=m;i>=1;i--)
      cout<<a[i]<<" ";
}

```

```

begin
  repeat
    readln(n,m);
    until((n>=1) and (n<=100) and (m>=1)
and (m<=n));

  for i:=1 to n do
    readln(a[i]);
end;
begin
  citire;
  ordonare(a,n);
  for i:=n-m+1 to n do
    write(a[i]:4:2,' ');
writeln;
  for i:=m downto 1 do
    write(a[i]:4:2,' ');
writeln;

end.

```

4. Explicații nu sunt necesare.

Varianta C/C++

```

#include<fstream.h>
ofstream f("sir.txt");
int nr;
void main()
{ //aranjamente de 26 luate cate 2
  for(char i='A';i<='Z'-1;i++)
    for(char j=i+1;j<='Z';j++)
      {f<<i<<j<<endl;
      }f<<j<<i<<endl;

f.close();
}

```

Varianta PASCAL

```

var f:text;
    nr:integer;
    i,j:byte;
begin
  assign(f,'sir.txt');
  rewrite(f);
  { //aranjamente de 26 luate cate 2}
  for i:=65 to 89 do
    begin
      for j:=i+1 to 90 do
        writeln(f,chr(i),chr(j));
        writeln(f,chr(j),chr(i));
      end;
    close(f);
  end.

```

Varianta <93>

SUBIECTUL I

1. b)

2.

a) 26

b)

Varianta C/C++

```

#include<fstream.h>
int n,m,s;
void main()
{cin>>n>>m;
  while(n<m)
  { s=s+n;
    n=n+3;
  }
  if(n==m)
    cout<<s+n;
  else
    cout<<0;
}

```

Varianta Pascal

```

var n,m,s:integer;
begin
  readln(n,m);
  while(n<m) do
  begin
    s:=s+n;
    n:=n+3;
  end;

  if(n=m) then
    writeln(s+n)

```

```

}
else
    writeln(0);
end.

```

c) valoarea lui m nu trebuie să fie de forma $3K+1$.

d) Explicați: Dacă m poate lua 6 valori distincte 2,3,5,6,8,9.

Se va afișa suma termenilor mai mici sau egali cu m , ai unei progresii aritmetice cu rația r și primul termen n .

```

citește n,m
    (numere naturale, n<=m)
s← [(m+n)/2]*[(m-n)/3+1];

dacă (m-n)%3=0
    scrie s;
altfel scrie 0;
sfarsit daca

```

SUBIECTUL II

1. d)

2. b)

3. 1 (trebuie să existe 4 noduri cu același tata)

4. Algoritmul elimină nodurile având numărul de ordine par (în lista inițială). Pentru o listă cu 99 de noduri vom elimina 49 (există 49 de numere pare între 1 și 99). După eliminare lista va avea 50 de noduri.

5.

Varianta C/C++

```

#include<fstream.h>
#include<string.h>
char s1[201],s2[201],voc[]="aeiou";
int vocala(char c)
{ if(strchr(voc,c)) return 1;
  else return 0;
}
void main()
{cin.get(s1,201);
 cin.get();

cin.get(s2,201);
for(int i=0;i<n;i++)
{int c1,c2;
 c1=vocala(s1[i]);
 c2=vocala(s2[i]);
 if(c1+c2==2) t[i]='*';
 else
     if(c1+c2==0) t[i]='#';
     else t[i]='?';
}
cout<<t<<endl;
}

```

Varianta PASCAL

```

var s1,s2,t:string[201];
    i,n,c1,c2:integer;
    voc:string;
function vocala(c:char):integer;
begin
    voc:='aeiou' ;
    if(pos(c,voc)=0) then
        vocala:=0
    else
        vocala:=1;
end;
begin
    readln(s1);
    readln(s2);
    t:=s1+t;
    n:=length(s1);
    for i:=1 to n do
        begin
            c1:=vocala(s1[i]);
            c2:=vocala(s2[i]);
            if(c1+c2=2) then
                t[i]:='*'
            else
                if(c1+c2=0) then
                    t[i]='#'
                else t[i]='?';
        end;
    end;
end.

```

```
end;
writeln(t);
end.
```

SUBIECTUL III

1. 4,4,4,3,3,3,3,3,2
2. 4,4,4,3,3,3,3,3,2
- 3.

Se deduce valoarea fiecărei componente pe baza poziției în cadrul vectorului.

Varianta C/C++

```
#include<fstream.h>
ofstream f("sir.txt");
int a[10000],n;
void main()
{ do
{cin>>n;
}while(n<0 || n%2==0);

a[1]=1;
for(int i=2;i<n;i=i+2)
{a[i+1]=i/2+1;
a[i]=n+1-i/2;
}
for(i=1;i<=n;i++)
cout<<a[i]<<" ";
cout<<endl;
}

O variantă care nu necesită
utilizarea unui vector este
cout<<endl;
for(int i=2;i<=n-1;i=i+2)
cout<<n+1-i/2<<" "<<(i+2)/2<<"
";
```

Varianta PASCAL

```
uses crt;
var a:array[1..10000] of integer;
n,i:integer;

begin
clrscr;
repeat
readln(n);
until(n>=0) and (n mod 2=1);

a[1]:=1;
i:=2;
while(i<=n-1) do
begin
a[i+1]:=i div 2+1;
a[i]:=n+1-i div 2;
i:=i+2;
end;

for i:=1 to n do
write(a[i], ' ');
writeln;
end.
```

4.

a) Vom căuta numărul cerut începând de la cel mai mare divizor comun al celor 2 numere a și b transmise ca parametru. Dacăg ăsim un divizor d al lui a și b atunci îl vom testa dacăse poate scrie ca produs de numere prime distincte. Pentru aceasta este suficient săg ăsim un divizor prim i pentru care i*i este deasemenea divizor(contraexemplu).

Dacănu e cîtă un astfel de divizor atunci funcția returnează 0.

```
int test(int d)
{for(int i=2;i<sqrt(d);i++)
if(d%i==0)
if(prim(i))
if(d%(i*i)==0) return
0;
return 1;
}
int dcp (int a,int b)
{ for(int d=cmmdc(a,b);d>=1;d--)
if(a%d==0 && b%d==0)
if (test(d)==1) return d;
return 1;
}
```

```
function test(d:integer):integer;
var i,rez:integer;
begin
rez:=0;
for i:=2 to trunc(sqrt(d)) do
if(d mod i=0) then
if(prim(i)=1) and (prim(d div
i)=1)and (i*i<>d) then
rez:=1;

test:=rez;
end;

function dcp(a,b:integer):integer;
```

```

var d, rez, gata: integer;
begin
  rez:=1;
  d:=cmmdc(a,b);
  gata:=0;
  while(d>=1) and (gata=0) do
    begin
      if(a mod d=0) and (b mod d=0)
    then
      if (test(d)=1) then
        begin
          rez:=d;
          gata:=1;
        end;
      d:=d-1;
    end;
  dcp:=rez;
end;

```

b) În variabila **d** preluăm, pe rând, divizorul cu proprietatea cerută, pentru fiecare pereche de două elemente formată de **d** și unul din elementele din vector.

Varianta C/C++

```

#include<fstream.h>
#include<math.h>
ifstream f("numere.in");
int a[101],n;
int prim(int x)
{ if(x<=1) return 0;
  if(x%2==0)
    if(x==2) return 1;
    else return 0;

  else for(int d=3;d*d<=x;d=d+2)
    if(x%d==0) return 0;
  return 1;
}
int cmmdc(int x,int y)
{ while(y)
  {int r=x%y;
   x=y;
   y=r;
  }
  return x;
}
int test(int d)
{for(int i=2;i<sqrt(d);i++)
  if(d%i==0)
    if(prim(i))
      if(d%(i*i)==0) return
0;
  return 1;
}
int dcp (int a,int b)
{ for(int d=cmmdc(a,b);d>=1;d--)
  if(a%d==0 && b%d==0)
    if (test(d)==1) return d;
} return 1;

void main()

```

Varianta PASCAL

```

var a:array[1..101] of integer;
    n,d,i,x:integer;
    f:text;

function prim(x:integer):integer;
var p,d:integer;
begin
  if(x<=1) then p:=0;
  if(x mod 2=0) then
    if(x=2) then p:=1
    else p:=0
  else
    begin
      d:=3;p:=1;
      while(d*d<=x) do
        begin
          if(x mod d=0) then p:=0;
          d:=d+2;
        end;
    end;
  prim:=p;
end;

function cmmdc(x,y:integer):integer;
var r:integer;
begin
  while(y>0) do
    begin
      r:=x mod y;
      x:=y;
      y:=r;
    end;
  cmmdc:=x;
end;

function test(d:integer):integer;
var i,rez:integer;
begin
  rez:=1;

```



```
{int d,x;
f>>n;
f>>x;
d=x;
for(int i=2;i<=n;i++)
    {f>>x;
    d=dcf(d,x);
    }
if (d==1) cout<<"Nu exista ";
else
    cout<<d<<endl;
}
```

```
for i:=2 to trunc(sqrt(d)) do
    if(d mod i=0) then
        if(prim(i)=1) then
            if(d mod (i*i)=0) then
                rez:=0;

test:=rez;
end;

function dcf(a,b:integer):integer;
var d,rez,gata:integer;
begin
    rez:=1;
    d:=cmmdc(a,b);
    gata:=0;
    while(d>=1) and (gata=0) do
        begin
            if(a mod d=0) and (b mod d=0)
then if (test(d)=1) then
                begin
                    rez:=d;
                    gata:=1;
                end;
            d:=d-1;
        end;
    dcf:=rez;
end;
begin
    assign(f,'numere.in');reset(f);
    readln(f,n);
    read(f,x);
    d:=x;
    for i:=2 to n do
        begin
            read(f,x);
            d:=dcf(d,x);
        end;
    if (d=1) then write('Nu exista ')
    else writeln(d);
end.
```

SUBIECTUL II

Varianta <93 neintensiv>

3. 1 (trebuie să existe 4 noduri cu același tată)
4. După inserare lista devine: 10,2,4,2,2,2,6. Secvența maximă este de lungime 3.

SUBIECTUL III

4.

Varianta C++

```
#include<fstream.h>
int n,d,x;
int cmmdc(int a, int b)
{ int r;
  while(b)
    {int r=a%b;
```

Varianta Pascal

```
var n,d,x,i:integer;
f:text;
function cmmdc(a,b:integer):integer;
var r:integer;
begin
  while(b>0) do
```

```

        a=b;
        b=r;
    }
    return a;
}
ifstream f("numere.in");
void main()
{ f>>n;
  f>>d;
  for(int i=2;i<=n;i++)
    { f>>x;
      d=cmmdc(d,x);
    }
  cout<<d<<endl;
}

```

```

begin
  r:=a mod b;
  a:=b;
  b:=r;
end;
cmmdc:=a;
end;

begin
assign(f,'numere.in');reset(f);
readln(f,n);
read(f,d);
for i:=2 to n do
  begin
    read(f,x);
    d:=cmmdc(d,x);
  end;
writeln(d);
end.

```

Varianta <94>

SUBIECTUL I

1. a)

2.

a) 621131

b) Dacă numărul conține cel puțin o cifră impar atunci aceasta va apărea la sfârșitul numărului x , deci nu poate fi multiplu de 10. Dacă numărul este format doar din cifre pare vom putea obține cel puțin un zero la sfârșitul numărului x , doar dacă numărul începe cu

0, ceea ce nu are sens. Deci numărul cerut este 0.

c) citește n (număr natural)

```

n1←0
n2←0
k1←0
cât timp n≠0 execută
  nr←[n/10]
  cf←n-nr*10
  dacă (cf=0 sau cf=2 sau cf=4 sau cf=6 sau cf=8 )
    n2←n2*10+cf
  altfel
    n1←n1*10+cf
    k1 k1+1
  n←[n/10]←
sfarsit cat timp
p←1
pentru i←1, k1 execută
  p←p*10
x←n2*p+n1
scrie x

```

d)

Varianta C/C++

```

#include<fstream.h>
long n,n1,n2,k1,p,x;
void main()
{ cin>>n;
  while(n)

```

Varianta Pascal

```

var n,n1,n2,k1,p,x:longint;
    i:integer;
begin
  readln(n);
  while(n>0) do

```

```

{ if(n%10%2==0)
    n2=n2*10+n%10;
  else
    {n1=n1*10+n%10;
      k1++;
    }
  n=n/10;
}
p=1;
for(int i=1;i<=k1;i++)
  p=p*10;
x=n2*p+n1;
cout<<x;
}

```

```

begin
  if(n mod 10 mod 2=0) then
    n2:=n2*10+n mod 10
  else
    begin
      n1:=n1*10+(n mod 10);
      k1:=k1+1;
    end;
  n:=n div 10;
end;
p:=1;
for i:=1to k1 do
  p:=p*10;
  x:=n2*p+n1;
writeln(x);
end.

```

SUBIECTUL II

1. b)

2. Dacă eliminăm muchiile (1,3) și (2,5) se obține un arbore. Vectorul de tați este 2 3 4 5 0

3.

1: 3
 2: 1,3
 3: 4,5
 4: 5
 5: 2

4. Varianta C/C++

```

#include<fstream.h>
#include<string.h>
char s[21],t[21];
int n;
void main()
{ cin.get(s,21);
  char *p=strchr(s,' ');
  n=p-s;
  while(p[1]!=' ')
    strcpy(p,p+1);
  strcpy(t,p+1);
  t[strlen(p+1)]= ' ';
  strncat(t,s,n);
  cout<<t<<endl;
}

```

5.

Varianta C/C++

```

char aux;
aux=prim->next->lit;
prim->next->lit=ultim->lit ;
ultim->lit=aux;

```

Varianta PASCAL

```

var s,t:string[21];
    n,p:integer;
begin
  readln(s);
  p:=pos(' ',s);
  while(s[p]=' ') do
    delete(s,p,1);

    t:=copy(s,p,length(s)-p+1);
    delete(s,p,length(s)-p+1);
    t:=t+' '+s;
  writeln(t);
end.

```

Varianta PASCAL

```

aux:char;
aux:=prim^.next^.lit;
prim^.next^.lit :=ultim^.lit;
ultim^.lit :=aux;

```

SUBIECTUL III

1. c)

2.

a) 68

b) Funcția calculează $2^{x+1}+4$. Valoarea maximă a lui x (număr natural) este 8.

3.

Vom extrage pe rând cifrele numărului a . Dacă există o cifră 0 sau mai mare decât n sau se repetă, atunci funcția returnează 0. De asemenea, dacă numărul cifrelor extrase din a diferă de n atunci funcția returnează 0. Altfel returnează 1.

Varianta C/C++

```
int verific(long a, int n)
{ int ap[10];
  for(int i=0;i<10;i++) ap[i]=0;
  int nr=0;
  while(a)
  { nr++;
    int cf=a%10;
    if(cf==0 || cf>n || ap[cf]==1)
      return 0;
    ap[cf]=1;
    a=a/10;
  }
  if(nr==n) return 1;
  else return 0;
}
```

Varianta PASCAL

```
function
verif(a:longint;n:integer):integer;
var ap:array[0..9] of integer;
    nr,i,cf,rez:integer;
begin
  for i:=0 to 9 do
    ap[i]:=0;
  nr:=0;
  rez:=1;
  while(a>0) do
  begin
    nr:=nr+1;
    cf:=a mod 10;
    if(cf=0) or (cf>n) or (ap[cf]=1)
    then
      rez:=0;
      ap[cf]:=1;
      a:=a div 10;
    end;
  if(nr=n) then verific:=rez
  else verific:=0;
end;
```

4. Explicații

a) Vom citi mai întâi numărul n din fișier, și numărul x de la tastatură. Apelăm funcția $verif(x)$. Dacă nu reprezintă o permutare a numerelor de la 1 la n atunci vom afișa mesajul "Nu apare". Altfel, vom citi pe rând câte o permutare din fișier pe care o vom testa dacă este egală cu x și apoi vom afișa numărul de ordine al liniei pe care aceasta apare.

Algoritmul este eficient din punct de vedere al memoriei deoarece nu sunt memorate toate permutările din fișier. Eficiența timp este dată de funcția $verif()$ care testează eficient dacă poate fi o permutare a valorilor de la 1 la n .

O altă posibilitate ar fi determinarea numărului de ordine al unei permutări date, în cadrul sirului ordonat de permutări dat utilizând o formulă. Dar în acest caz nu se mai justifică conținutul fișierului de intrare (nu mai avem nevoie de permutări ci doar de n).

b)

Varianta C/C++

```
#include<fstream.h>
#include<string.h>
#include<stdlib.h>

ifstream f("permut.txt");
char s1[10],sir[10];
long x;
int n;

int verific(long a, int n)
{ int ap[10];
  for(int i=0;i<10;i++) ap[i]=0;
```

Varianta PASCAL

```
var s1,sir:string;
    x:longint;
    n:integer;
    f:text;

function
verif(a:longint;n:integer):integer;
var ap:array[0..9] of integer;
    nr,i,cf,rez:integer;
begin
  for i:=0 to 9 do
    ap[i]:=0;
```

```

int nr=0;
while(a)
{ nr++;
  int cf=a%10;
  if(cf==0 || cf>n ||
ap[cf]==1) return 0;
  ap[cf]=1;
  a=a/10;
}
if(nr==n) return 1;
else return 0;
}
long linia(long x)
{ long l=1,y;
  while(f>>y)
  { l++;
    if(x==y)
      return l;
  }
  return 1;
}
void main()
{
  do
  {cin>>x;
  }while(x<=0);
  f>>n;

  if(verif(x,n)==0) cout<<"NU
apare!";
  else
    cout<<"apare pe linia
"<<linia(x);
}

```

```

nr:=0;
rez:=1;
while(a>0) do
begin
  nr:=nr+1;;
  cf:=a mod 10;
  if(cf=0) or (cf>n) or (ap[cf]=1)
then
  rez:=0;
  ap[cf]:=1;
  a:=a div 10;
end;
if(nr=n) then verific:=rez
else verific:=0;
end;

function linia(x:longint):longint;
var rez,l,y:longint;

begin
  while(not eof(f)) do
  begin
    read(f,y);
    l:=l+1;;
    if(x=y) then
      rez:=l;
    end;
    linia:=rez;
  end;

begin
assign(f,'permut.txt');reset(f);
repeat
  readln(x);
until(x>0);
readln(f,n);

if(verif(x,n)=0) then writeln('NU
apare!')
else
  writeln('apare pe linia ',linia(x));
end.

```

Varianta <94 neintensiv>

SUBIECTUL II

2. a)

3. Dacă eliminăm muchiile (1,3) și (2,5) se obține un arbore. Vectorul de tați este 2 3 4 5 0

- 4.
- 1: 3
 - 2: 1,3
 - 3: 4,5
 - 4: 5
 - 5: 2

5.

Varianta C/C++

```
#include<fstream.h>
```

Varianta PASCAL

```
var s,t:string[21];
```

```
#include<string.h>
char s[21],t[21];
int n;
void main()
{ cin.get(s,21);
  char *p=strchr(s,' ');
  n=p-s;
  while(p[1]==' ')
    strcpy(p,p+1);
  strcpy(t,p+1);
  t[strlen(p+1)]=' ';
  strncat(t,s,n);
  cout<<t<<endl;
}
```

```
n,p:integer;
begin
  readln(s);
  p:=pos(' ',s);
  while(s[p]=' ') do
    delete(s,p,1);

  t:=copy(s,p,length(s)-p+1);
  delete(s,p,length(s)-p+1);
  t:=t+' '+s;
  writeln(t);
end.
```

SUBIECTUL III

3.

a) Varianta C/C++

```
int f(int n,int a[201],int
p1,int p2)
{ int neg=0,poz=0,nul=0;
  for(int i=p1;i<=p2;i++)
    if(a[i]<0)neg++;
    else
      if(a[i]==0) nul=0;
      else poz++;
  if(neg<poz)
    if(poz<nul) return 0;
    else
      if(poz==nul) return 2;
      else return 1;
  else
    if(neg>poz)
      if(neg<nul) return 0;
      else
        if(neg==nul) return 2;
        else return -1;
    else
      return 2
}
```

b)

```
#include<fstream.h>
int n,a[201],i;
void main()
{cin>>n;
for(i=0;i<n;i++) cin>>a[i];

int rez=f(n,a,0,n-1);
```

Varianta PASCAL

```
function
f(n:integer;a:vector;p1,p2:integer):integer;
var neg,poz,nul,i:integer;
begin
  neg:=0;
  poz:=0;
  nul:=0;
  for i:=p1 to p2 do
    if(a[i]<0) then
      neg:=neg+1
    else
      if(a[i]=0) then
        nul:=0
      else poz:=poz+1;
  if(neg<poz) then
    if(poz<nul) then
      f:=0
    else
      if(poz=nul) then f:=2;
      else f:=1
  else
    if(neg>poz)
      if(neg<nul) then

        else f:=0
        if(neg>nul) then f:=-1;
        else f:=2;
  else
    if(nul>poz) then f:=0
    else f:=2;
end;
type vector=array[1..201] of integer;
var n,i,rez:integer;
  a:vector;
function
f(n:integer;a:vector;p1,p2:integer):integer;
var neg,poz,nul,i:integer;
begin
  neg:=0;
  poz:=0;
```

```

if(rez==-1) cout<<"Negative";
else
    if(rez==0) cout<<"Nule";
    else cout<<"Pozitive";
}

nul:=0;
for i:=p1 to p2 do
    if(a[i]<0) then
        neg:=neg+1
    else
        if(a[i]=0) then
            nul:=0
        else poz:=poz+1;
if(neg<poz) then
    if(poz<nul) then
        f:=0
    else f:=1
else
    if(neg<nul) then
        f:=0
    else f:=-1;
end;

begin
    readln(n);
    for i:=0 to n-1 do
        read(a[i]);

    rez:=f(n,a,0,n-1);
    if(rez=-1) then
        write('Negative')
    else
        if(rez=0) then
            write('Nule')
        else
            write('Pozitive');
end.

```

4. Explicații

a) Vom citi pe rând numere din fișier. Vom determina diferența minimă în modul și pentru fiecare actualizare a minimumului vom reține și valoarea care a minimizat diferența.

Eficiența ca memorie utilizată provine din faptul că nu sunt memorate numerele citite într-un vector, ci este utilizată o singură variabilă.

b)

Varianta C/C++

```

#include<fstream.h>
#include<math.h>
ifstream f("bac.txt");

float x,max;
long n,nr;
void main()
{ cin>>x;
  f>>n;
  max=fabs(x-n);
  nr=n;
  while(f>>n)
      if(fabs(x-n)<max)
          {max=fabs(x-
n);
          nr=n;

  cout<<"Numarul cel mai
apropiat este "<<nr;
}

```

Varianta PASCAL

```

var x,max:real;
    n,nr:longint;
    f:text;

begin
    readln(x);
    assign(f,'bac.txt');reset(f);
    readln(f,n);
    max:=abs(x-n);
    nr:=n;
    while(not eof(f)) do
        begin
            read(f,x);
            if(abs(x-n)<max) then
                begin
                    max:=abs(x-n);
                    nr:=n;
                end;
        end;
    write('Numarul cel mai apropiat este ',nr);
end.

```

| end.

Varianta <95>

SUBIECTUL I

1. d) Algoritmul construiește oglinditul numărului n format doar din cifrele (strict) mai mici decât k.

- a) 125
- b) 98002, 80029, 80092, 89002, 80902,

c)

Varianta C/C++

```
#include<fstream.h>
long n, k, x;
void main()
{
    cin>>n>>k;
    while(n)
    {
        if(n%10<k)
            x=x*10+n%10;
        n=n/10;
    }
    cout<<x;
}
```

Varianta Pascal

```
var n, k, x: longint;
begin
    readln(n, k);

    while(n>0) do
        begin
            if(n mod 10<k) then
                x:=x*10+n mod 10;
            n:=n div 10;
        end;
    writeln(x);
end.
```

d)

```
citește n, k (numere naturale nenule)
x←0;
repetă
    dacă n%10<k atunci
        x←x*10+n%10;
    n←[n/10]
până când n=0
scrie x
```

SUBIECTUL II

- 1. b)
- 2. d)

3. Deoarece graful complet K_5 este eulerian, prin eliminarea oricărei muchii obținem un graf neeulerian cu număr maxim de muchii. Dacă eliminăm muchia (4,5) listele de adiacență vor fi:

- 1: 2,3,4,5
- 2: 1,3,4,5
- 3: 1,2,4,5
- 4: 1,2,3
- 5: 1,2,3

4. Lungimea maximă a drumului de la 1 la 5 este 5.
 Muchiile folosite sunt (1,2),(2,4),(4,3),(3,2),(2,5).

5.

Varianta C/C++

```
#include<fstream.h>
#include<string.h>
char s[21], t[21];
320
```

Varianta PASCAL

```
var s, t: string[21];
    i, p, n: integer;
begin
```



```
int i;
void main()
{ cin.get(s,21);
  char *p=strchr(s, ' ');
  i=p-s;
  while(s[i]!=' ') i++;
  t[0]=s[i];
  t[1]='.';
  t[2]=' ';
  strncat(t,s,p-s) ;
  cout<<strlwr(t);
}
```

```
readln(s);
t:=' ';
n:=length(s);
p:=pos(' ',s);
while(s[p+1]=' ') do
  delete(s,p+1,1);
t[1]:=s[p+1];
t[2]:= '.';
t[3]:= ' ';
delete(s,p,n-p);
t:=t+s;
writeln(t);
end.
```

SUBIECTUL III

1. b)

2. Pentru 8,11,14,17,22 se vor returna valorile 0,1,2,3 respectiv 4.

3. Varianta C/C++

```
int f(int n,int a[201])
{ int nrpoz,nrneg;
  nrpoz=nrneg=0;
  for(int i=1;i<=n;i++)
    if(a[i]>0) nrpoz++;
    else
      if(a[i]<0) nrneg++;

  if(nrneg>nrpoz) return -1;
  else
    if(nrneg==nrpoz) return 0;
    else return 1;
}
```

Varianta PASCAL

```
type vector=array[1..201] of integer;
function
f(n:integer;a:vector):integer;
var nrpoz,nrneg,i:integer;
begin
  nrpoz:=0;
  nrneg:=0;
  for i:=1 to n do
    if(a[i]>0) then
      nrpoz:=nrpoz+1
    else
      if(a[i]<0) then
        nrneg:=nrneg+1;
  if(nrneg>nrpoz) then
    f:=-1
  else
    if(nrneg=nrpoz) then
      f:=0
    else f:=1;
end;
```

4.

a)

Varianta C/C++

```
#include <fstream.h>
ofstream f("rez.dat");
long s,x,y,a,b,c;
void main()
{cin>>s;
  long max=1;
  for(x=1;x<=s/3;x++)
    for(y=x;y<=(s-x)/2;y++)
      {long z=s-x-y;
        if(x*y*z>max)
          {max=x*y*z;
            a=x;
            b=y;
            c=z;
          }
      }
}
```

Varianta PASCAL

```
var s,x,y,z,a,b,c,max:longint;
f:text;
begin
  assign(f,'rez.dat');
  rewrite(f);

  readln(s);
  max:=1;
  for x:=1 to s div 3 do
    for y:=x to (s-x) div 2 do
      begin
        z:=s-x-y;
        if(x*y*z>max) then
          begin
            max:=x*y*z;
            a:=x;

```

```

f<<a<<" "<<b<<" "<<c<<endl;
f.close();
}
|
b:=y;
c:=z;
end;
end;
writeln(f,a,' ',b,' ',c);
close(f);
end.

```

b) Pentru eficiență din punct de vedere al timpului de execuție vom impune câteva limitări pentru valorile posibile ale valorilor x, y, z . Astfel, cele 3 numere vor fi în ordine crescătoare pentru a elimina comparații inutile. Obținem pentru cel mai mic număr relația $3 \cdot x \leq S$ și pentru cel mai mare, $z = S - x - y \geq y$ de aici $y \in [x, (S-x)/2]$. Pentru fiecare triplet astfel construit actualizăm variabila \max și dacă este cazul reținem în variabilele globale a, b, c numerele determinate.

Varianta <95 neintensiv>

SUBIECTUL II

3. Lungimea maximă a drumului de la 1 la 5 este 5.
Muchiile folosite sunt (1,2),(2,4),(4,3),(3,2),(2,5).
4.
 - 1 : 2
 - 2 : 1
 - 3 : 4
 - 4 : 3
 - 5 : 6, 7, 8
 - 6 : 5, 7, 8
 - 7 : 5, 6, 8
 - 8 : 5, 6, 7

SUBIECTUL III

2.
 - a) 0
 - b) 3

Varianta <96>

SUBIECTUL I

1. c)
- 2.

a) Algoritmul afișează toate descompunerile distincte ale lui n ca sumă de numere naturale distincte. Pentru $n=10$ valorile afișate vor fi :

```

1 2 7
1 3 6
1 4 5
2 3 5

```

- b) $12 = 3 + 4 + 5$
- c)

Varianta C/C++

```

#include<iostream.h>
int n,i,j,k;
void main()
{ cin>>n;

```

Varianta Pascal

```

var n,i,j,k:integer;
begin
  readln(n);
  for i:=1 to n do
    for j:=1 to n do

```

```

for(i=1;i<=n;i++)
for(j=1;j<=n;j++)
for(k=1;k<=n;k++)
if(i<j && j<k)
if(i+j+k==n)
{cout<<i<<" "<<j<<" "<<k;
cout<<endl;
}
}
}

```

```

for k:=1 to n do
begin
if(i<j) and (j<k) then
if(i+j+k=n) then
writeln(i, ' ', j, ' ', k);
end;
end.

```

d) O variantă eficientă a timp de execuție se poate obține impunând restricții pentru i și j astfel încât să poată fi eliminate cele două if-uri din program.

```

Citește n (număr natural)
Pentru i←1, n/3 execută
Pentru j←i+1, (n-i-1)/2 execută
Scrie i, ' ', j, ' ', n-i-j
Salt la rând nou

```

SUBIECTUL II

1. a) cel mult 9 valori egale

2. a)

3. Putem considera cazul grafului format dintr-un ciclu elementar (deci este hamiltonian) la care mai adăugăm o muchie oarecare, de exemplu (2,5), graful obținut nefiind eulerian.

Listele de adiacență sunt :

```

1: 2,5
2: 1,3,5
3: 2,4
4: 3,5
5: 1,2,4

```

4. (1,2), (2,4), (4,3), (3,2), (2,5)

5.

Varianta C/C++

```

#include<fstream.h>
#include<string.h>
char s1[21], s2[21], t[41];
void main()
{cin.get(s1, 21);
cin.get();

cin.get(s2, 21);
strcpy(t, s2);
t[strlen(s2)]=' ';
strcat(t, s1);
t[strlen(t)]='\0';
cout<<t;
}

```

Varianta PASCAL

```

var s1, s2, t:string[41];
begin
readln(s1);
readln(s2);

t:=t+s2+' '+s1;
writeln(t);
end.

```

SUBIECTUL III

1. b)

2. 10, 13, 16, 18

3.

a)

Varianta C/C++

```

int verific(int a, int b, int c)

```

Varianta PASCAL

```

function

```

```
{ return ((a+b>c) && (a+c>b) && (b+c>a));
}
```

```
verif(a,b,c:integer):integer;
begin
  if (a+b>c) and (a+c>b) and (b+c>a)
  then verific:=1
  else verific:=0;
end;
```

b) Dacă există acele două triunghiuri și dacă au laturile corespunzătoare congruente atunci triunghiurile sunt congruente (cazul L.L.L.). Pentru aceasta vom compara lungimile laturilor în ordine corespunzătoare (cea mai mică din primul triunghi cu cea mai mică din al doilea triunghi, etc.).

Varianta C/C++

```
#include<fstream.h>
ifstream ff("numere.in");
int a,b,c,d,e,f;
int verific(int a,int b,int c)
{ return ((a+b>c) && (a+c>b) && (b+c>a));
}

void sch(int &x,int &y)
{int aux=x;x=y;y=aux;}

void ordonare(int &x,int &y,int &z)
{ if(x>y) sch(x,y);
  if(x>z) sch(x,z);
  if(y>z) sch(y,z);
}

void main()
{ cin>>a>>b>>c>>d>>e>>f ;
  if(verific(a,b,c) &&
verific(d,e,f))
  { ordonare(a,b,c);
    ordonare(d,e,f);
    if(a==d && b==e &&
c==f)

    cout<<"congruente";
    else
    cout<<"necongruente";
  }
  else
  cout<<"NU";
}
```

Varianta PASCAL

```
var a,b,c,d,e,f:integer;
function
verif(a,b,c:integer):integer;
begin
  if(a+b>c) and (a+c>b) and
(b+c>a) then

    verific:=1
  else verific:=0;
end;
procedure sch(var x,y:integer);
var aux:integer;
begin
  aux:=x;x:=y;y:=aux;
end;

procedure ordonare(var
x,y,z:integer);
begin
  if(x>y) then sch(x,y);
  if(x>z) then sch(x,z);
  if(y>z) then sch(y,z);
end;
begin
  readln(a,b,c,d,e,f);
  if(verific(a,b,c)=1) and
(verific(d,e,f)=1) then
  begin
    ordonare(a,b,c);
    ordonare(d,e,f);
    if(a=d) and (b=e) and (c=f)
then

    else writeln('congruente')
    writeln('necongruente');
  end
  else
  writeln('NU');
end.
```

4. Explicații

a) Vom citi din fișier n și m de pe prima linie a fișierului și vom memora și elementele de pe a doua linie a fișierului în vectorul x . A treia linie va fi doar parcursă pentru a determina elementele vecine cerute. Pentru fiecare pereche de elemente de pe a treia linie vom determina în vectorul x câte elemente cuprinse între acestea există, actualizând de fiecare dată variabila max și preluând, dacă este cazul, în variabilele A și B capetele noului

interval găsit.

Eficiența ca timp de execuție se datorează unei singure parcurgeri a vectorului x și a ultimei linii a fișierului. Acest lucru este posibil deoarece se precizează în enunț că elementele sunt ordonate strict crescător. (rezultă că fiecare element x[i] se poate afla în interiorul unui singur interval).

Eficiența ca spațiu de memorie utilizat rezultă din faptul că pentru al doilea șir nu utilizăm decât o variabilă (nu este necesară memorarea acestuia).

b)

Varianta C/C++

```
#include<fstream.h>
int x[1001],i,n,m,a,b,max,nr,A,B;
ifstream f("date.in");
void main()
{
    f>>n>>m;

    for(int i=1;i<=n;i++)
        f>>x[i];

    f>>a;
    i=1;
    while(x[i]<a)
        i++;

    while(f>>b)
    { //caut nr din x intre a si b
        nr=0;
        if(a==x[i-1]) nr=1;
        while(x[i]<=b && i<=n)
            { i++;
              nr++;
            }

        if(nr>=max)
            {max=nr;
              A=a;
              B=b;
            }
        a=b;
    }

    cout<<endl<<"["<<A<<","<<B<<"]"<<endl<<max<<endl<<"numere";
}
```

Varianta PASCAL

```
var f:text;
var x:array[1..1001] of integer;
    i,n,m,a,b,max,nr,aa,bb:integer;
begin
    assign(f,'bac.dat');reset(f);
    readln(f,n,m);

    for i:=1 to n do
        read(f,x[i]);

    readln(f);
    read(f,a);
    i:=1;
    while(x[i]<a) do
        i:=i+1;

    while(not eof(f)) do
        begin
            read(f,b);
            {caut nr din x intre a si b}
            nr:=0;
            if(a=x[i-1]) then
                nr:=1;
            while(x[i]<=b) and (i<=n) do
                begin
                    i:=i+1;
                    nr:=nr+1;
                end;

            if(nr>=max) then
                begin
                    max:=nr;
                    aa:=a;
                    bb:=b;
                end;
            a:=b;
        end;

    writeln(['[',aa,',',bb,']',' ',max,' numere');
end.
```

Varianta <96 neintensiv>**SUBIECTUL II**

3. (1,2), (2,4), (4,3), (3,2), (2,5)

4. 1: 2 ; 2: 1 ; 3: 4 ; 4: 3 ; 5: 6 ; 6: 5 ; 7: 8 ; 8: 7

SUBIECTUL III**4. Explicații**

a) Vom citi din fișier n și m de pe prima linie a fișierului și pentru elementele de pe a doua linie a fișierului vom determina cea mai mică respectiv cea mai mare valoare în variabilele min și max fără a memora elementele citite. Următoarele m linii vor fi doar parcurse pentru a determina câte din cele m perechi de numere citite (intervale) pot conține toate elementele de pe linia a doua, comparând capetele intervalului citit cu minimul și maximumul determinat.

b)

Varianta C/C++

```
#include<fstream.h>
ifstream f("bac.dat");
int min,max,x,n,m,nr,a,b;
void main()
{f>>n>>m;
 f>>x;
 min=max=x;
 for(int i=2;i<=n;i++)
 {f>>x;
  if(min>x) min=x;
  if(max<x) max=x;
 }
 cout<<min<<" "<<max<<endl;
 for(i=1;i<=m;i++)
 {f>>a>>b;
  if(a>b)
   {int aux=a;a=b;b=aux;}
  if((a<=min) && (max<=b))
   nr++;
 }
 cout<<"Numarul de intervale este
 "<<nr;
 }
```

Varianta PASCAL

```
var
min,max,aux,x,n,m,nr,a,b,i:integer;
 f:text;
begin
 assign(f,'bac.dat');
 reset(f);

 readln(f,n,m);
 read(f,x);
 min:=x;
 max:=x;

 for i:=2 to n do
 begin
 read(f,x);
 if(min>x) then
 min:=x;
 if(max<x) then
 max:=x;
 end;
 writeln(min,' ',max);
 readln(f);
 for i:=1 to m do
 begin
 read(f,a,b);
 if(a>b) then
 begin
 aux:=a;
 a:=b;
 b:=aux;
 end;
 if(a<=min) and (max<=b) then
 nr:=nr+1;
 end;
 writeln('Numarul de intervale este
 ',nr);
 end.
```

SUBIECTUL I

1. b)

2.

b) 3, 1, 2, 3, 4

c) citește x,y,z
 scrie x%10
 scrie 2*(y%10)
 scrie 3*(z%10)

d)

Varianta C/C++

```
#include<fstream.h>
int n,i,j;
long x;
void main()
{
for(i=1;i<=3;i++)
{cin>>x;
int s=0;
for(j=1;j<=i;j++)
s=s+x%10;
cout<<s<<endl;
}
}
```

Varianta Pascal

```
var n,i,j:integer;
s,x:longint;
begin
for i:=1 to 3 do
begin
readln(x);
s:=0;
for j:=1 to i do
s:=s+x mod 10;
writeln(s);
end;
end.
```

SUBIECTUL II

1. c)

2. a)

3. TITA

4.

Varianta C/C++

```
typedef struct Nod
{int info;
Nod *adr;
} *Pnod;
Pnod p;
cout<<p->adr->adr->info;
```

Varianta PASCAL

```
type Pnod=^Nod;
Nod=record
info:integer;
adr:PNod;
end;
var p:Pnod;
write(p^.adr^.adr^.info);
```

5. Explicații

Varianta C/C++

```
#include<fstream.h>
#include<string.h>
int a[51][51],n,m,i,j;
void inversare_coloana(int j)
{for(int i=1;i<=n/2;i++)
{aux=a[i][j];
a[i][j]=a[n-i+1][j];
```

Varianta PASCAL

```
var a:array[1..51,1..51] of integer;
n,m,i,j:integer;
procedure
inversare_coloana(j:integer);
var aux,i:integer;
begin
for i:=1 to n div 2 do
begin
```

```

        a[n-i+1][j]=aux;
    }
}
void citire()
{ cin>>n>>m;
  for(i=1;i<=n;i++)
    for(j=1;j<=m;j++)
      cin>>a[i][j];
}
void afisare()
{ for(i=1;i<=n;i++)
  { for(j=1;j<=m;j++)
    cout<<a[i][j]<<" ";
    cout<<endl;
  }
}
void main()
{citire();

  for(i=1;i<=n;i++)
    inversare_coloana(j);
  afisare();
}

```

```

        aux:=a[i][j];
        a[i][j]:=a[n-i+1][j];
        a[n-i+1][j]:=aux;
    end;
end;
procedure citire();
begin
  writeln(n,m);
  for i:=1 to n do
    for j:=1 to m do
      read(a[i][j]);
    end;
  procedure afisare();
  begin
    for i:=1 to n do
      begin
        for j:=1 to m do
          write(a[i][j], ' ');
        writeln;
      end;
    end;
  begin
    citire();
    for j:=1 to n do
      inversare_coloana(j);
      afisare();
    end.
  end.
end.

```

SUBIECTUL III

1. b)

2. agc, agf, agg

a)

Varianta C/C++

```
void calcul(int k,int &s)
```

b) Suma componentelor pozitive din vector, aflate între pozițiile x și y inclusiv, se obține prin diferența sumei valorilor pozitive de la poziția x până la n și suma celor pozitive de la poziția y+1 până la n.

Varianta C/C++

```

#include<fstream.h>
int v[51],n;
void calcul(int k,int &s)
{ s=0;
  for(int i=k;i<=n;i++)
    if(v[i]>=0)s=s+v[i];
}

```

```

void main()
{int x,y;
  cin>>n;
  for(int i=1;i<=n;i++)
    cin>>v[i];

```

```

  cin>>x>>y;
  int s1,s2;
  calcul(x,s1);
  calcul(y+1,s2);

```

Varianta PASCAL

Program

Varianta PASCAL

```

var v:array[1..51] of integer;
    n,x,y,i:integer;
    s1,s2:integer;
procedure calcul(k:integer;var
s:integer);
var i:integer;
begin
  s:=0;
  for i:=k to n do
    if(v[i]>=0) then
      s:=s+v[i];
  end;
begin
  readln(n);
  for i:=1 to n do
    readln(v[i]);

```



```
cout<<"Suma ceruta este "<<s1-s2;
}
```

```
readln(x,y);
calcul(x,s1);
calcul(y+1,s2);
writeln('Suma ceruta este ',s1-s2);
end.
```

4. Explicații

a)

Varianta C/C++

```
#include<fstream.h>
#include<math.h>
ifstream f("date.txt");
long k,x,y;
float z;
void main()
{
    f>>k;
    for(x=1;x<=sqrt(k/2);x++)
        {z=sqrt(k-x*x);
        if(z==long(z))
            cout<<x<<" "<<z<<endl;
        }
}
```

Varianta PASCAL

```
Var k,x,y:longint;
    z:real;
    f:text;
begin
    assign(f,'date.txt');
    reset(f);
    readln(f,k);
    for x:=1 to trunc(sqrt(k div 2))
do
    begin
        z:=sqrt(k-x*x);
        if(z=trunc(z)) then
            writeln(x,' ',trunc(z));
    end;
end.
```

b) Pentru a obține un algoritm eficient ca timp de execuție, vom impune câteva limitări pentru valorile posibile ale lui x și y . Din faptul că $x \leq y$ și $x^2 + y^2 = k$ rezultă că este mai mic decât radicalul din $(k/2)$. Dacă diferența $k - x^2$ este pătrat perfect atunci avem o valoare bună pentru y , adică am găsit o soluție. Eficiența constă în faptul că valorile pentru y sunt deduse pe baza valorilor lui x , deci nu este necesară o parcurgere pentru y .

Varianta <97 neintensiv>

SUBIECTUL II

4.

```
POP          : 1,3
PUSH 4       : 1,3,4
PUSH 6       : 1,3,4,6
POP          : 1,3,4
```

SUBIECTUL III

1. b)

2. agc, agf, agg

3.

Varianta C/C++

```
void suma(int x, int y, long &z)
{z=0;
 int i=x;
 if(i%2==1) i++;
 for(;i<=y;i=i+2)
     z=z+i;
}
```

Varianta PASCAL

```
procedure suma(x,y:integer; var
z:long);
var i:integer;
begin
    z:=0;
    i:=x;
    if(i mod 2=1) then
        i:=i+1;

    while(i<=y) do
begin
        z:=z+i;
        i:=i+2;
```

4.

a)

Varianta C/C++

```
#include<fstream.h>
ifstream f("date.txt");
int a[10001],b[10001],p[10];
int n,i;
void main()
{f>>n;
  for(i=1;i<=n;i++)
    {f>>a[i];
      p[a[i]%10]++;
    }
  for(i=1;i<=9;i++)
    p[i]=p[i]+p[i-1];

  for(i=1;i<=n;i++)
    {b[p[a[i]%10]]=a[i];
      p[a[i]%10]--;
    }
  for(i=1;i<=n;i++)
    cout<<b[i]<<" ";
}
```

```
end;
end;
```

Varianta PASCAL

```
var a,b:array[1..10001] of integer;
    p:array[0..10] of integer;
    f:text;
    n,i:integer;
begin
  assign(f,'date.txt');
  reset(f);
  readln(f,n);
  for i:=1 to n do
    begin
      read(f,a[i]);
      Inc(p[a[i] mod 10]);
    end;
  for i:=1 to 9 do
    p[i]:=p[i]+p[i-1];

  for i:=1 to n do
    begin
      b[p[a[i] mod 10]]:=a[i];
      Dec(p[a[i] mod 10]);
    end;

  for i:=1 to n do
    write(b[i],' ');
  end.
```

b) Pentru a obține eficientă ca timp de execuție nu vom utiliza nici un algoritm de sortare, ci vom număra câte numere având aceeași cifră există a (principiul sortării prin numărare).

Vom construi în vectorul p numărul de elemente care se termină cu fiecare cifră și apoi transformăm vectorul p astfel încât p[i] să reprezinte poziția unde trebuie așezat un element cu ultima cifră i. După așezarea unui element vom decremența valoarea p[i] pentru a avea poziția următorului element cu ultima cifră i. Elementele sunt așezate pe locurile finale în vectorul b și sunt apoi afișate. Algoritmul este liniar.

Varianta <98>**SUBIECTUL I**

1. d)

2.

a) 3

b) Algoritmul afișează cel mai mare număr natural mai mic sau egal cu radicalul din n. Pentru a se afișa 4 putem introduce orice valoare naturală din intervalul [16,24].

c) citește n (număr natural)

scrie $\lfloor \sqrt{n} \rfloor$

d)

Varianta C/C++

```
#include<fstream.h>
long i,n;
void main()
{ cin>>n;
  while(i*i<=n) i++;
```

Varianta Pascal

```
var i,n:integer;
begin
  readln(n);
  i:=0;
  while(i*i<=n) do
```

```

cout<<i-1;
}
i:=i+1;
writeln(i-1);
end.

```

SUBIECTUL II

1. b)
2. a)

3. Vectorul nu poate avea decât 6 noduri. Vectorul de tați este : 0 1 1 2 3 3

4.

Varianta C/C++

```

struct COLET
{ float cost;
  float masa;
  char expeditor[31];
};
COLET x;

cin>>x.cost;
cin>>x.masa;
cin.get();
cin.get(x.expeditor,31);

```

Varianta PASCAL

```

type COLET=record
  cost:real;
  masa:real;
  expeditor:string[31];
end;
var x:COLET;
begin
  readln(x.cost);
  readln(x.masa);

  readln(x.expeditor);
  {writeln(x.cost:4:2,' ',x.masa:4:2,'
  ',x.expeditor);}
end.

```

5.

Varianta C/C++

```

#include<fstream.h>
int a[51][51],i,j,n,k;
void main()
{ cin>>n;
  for(i=1;i<=n;i++)
    for(j=1;j<=n;j++)
      {a[i][j]=2*k;
       k++;
      }
  for(i=1;i<=n;i++)
    {for(j=1;j<=n;j++)
      cout<<a[i][j]<<" ";
      cout<<endl;
    }
}

```

Varianta PASCAL

```

var a:array[1..51,1..51] of integer;
    i,j,n,k:integer;
begin
  readln(n);
  for i:=1 to n do
    for j:=1 to n do
      begin
        a[i][j]:=2*k;
        k:=k+1;
      end;
    for i:=1 to n do
      begin
        for j:=1 to n do
          write(a[i][j],' ');
        writeln;
      end;
    end;
end;

```

SUBIECTUL III

1. c)

2. {2} ; {2,5} ; {5}

3. Esența acestei probleme constă în rezolvarea următoarei inecuații de clasa a X-a: $x^k < y$ și alegerea valorii naturale maxime a lui x . Ecuația este echivalentă cu $k < \log_x y$ și de aici k maxim este egal cu partea întreagă a logaritmului. Pentru situațiile în care x este 0 sau 1, precum și multe altele, problema nu are sens. Aceste cazuri au fost prezentate în cadrul funcției **număr**, caz în care am returnat valoarea -1 (cu sens de eroare).

Varianta C/C++

```

long numar(int x, long y)
{ //daca x =1 si y=0 sau 1 problema

```

Varianta PASCAL

```

function
numar(x:integer;y:longint):longint;

```

```

nu are solutie
  //daca x>=1 si y=0 problema nu
  are solutie
  //daca x=y=0 problema deasemeni
  nu are sens
  if(x==0)
    if(y>=1) return 0;
    else return -1; //eroare
  else
    if(x==1 )
      if(y>=2) return 1;
      else return -1;//eroare
    else
      if(y==0) return -1; //eroare
      else
        return long(log(y)/log(x));
}

```

4.

a)

Varianta C/C++

```

#include<fstream.h>

long a[3001],b[3001],n,m,i,j,k1,k2,x;
ifstream f("date.txt");
int par(long x)
{ return (x%2==0); }
void main ()
{ f>>n>>m;
  for(i=1;i<=n;i++)
    {f>>x;
     if (par(x))a[++k1]=x;
    }
  for(i=1;i<=m;i++)
    { f>>x;
     if(par(x)) b[++k2]=x;
    }
  i=1;
  j=k2;
  while(i<=k1 && j>=1)
    if(a[i]<b[j])
      { cout<<a[i]<<" ";
        i++;
      }
    else
      { cout<<b[j]<<" ";
        j--;
      }
  while(i<=k1)
    { cout<<a[i]<<" ";
      i++;
    }
}

```

```

var rez:integer;
begin
  {daca x =1 si y=0 sau 1 problema nu
  are solutie
  daca x>=1 si y=0 problema nu are
  solutie
  daca x=y=0 problema deasemeni nu
  are sens
  }
  if(x=0) then
    if(y>=1) then
      rez:=0
    else
      rez:=-1 {eroare}
  else
    if(x=1) then
      if(y>=2) then
        rez:=1
      else
        rez:=-1 {eroare}
    if(y=0) then
      rez:=-1 {eroare}
    else
      rez:= trunc(ln(y)/ln(x));
  numar:=rez;
end;

```

Varianta PASCAL

```

var f:text;
    a,b:array[1..3001] of integer;
    n,m,i,j,k1,k2,x:integer;
function par(x:longint):integer;
begin
  if (x mod 2=0) then
    par:=1
  else par:=0;
end;
begin
  assign(f,'date.txt');
  reset(f);
  readln(f,m,n);
  for i:=1 to n do
    begin
      read(f,x);
      if (par(x)=1) then
        begin
          k1:=k1+1;
          a[k1]:=x;
        end;
    end;
  for i:=1 to m do
    begin
      read(f,x);
      if(par(x)=1) then
        begin
          k2:=k2+1;
          b[k2]:=x;
        end;
    end;

```

```

    }
while (j>=1)
    { cout<<b[j]<<" ";
      j--;
    }
}

```

```

end;
end;
i:=1;
j:=k2;
while(i<=k1) and (j>=1) do
    if(a[i]<b[j]) then
        begin
            write(a[i], ' ');
            i:=i+1;
        end
    else
        begin
            write(b[j], ' ');
            j:=j-1;
        end;
    while(i<=k1) do
        begin
            write(a[i], ' ');
            i:=i+1;
        end;
    while(j>=1) do
        begin
            write(b[j], ' ');
            j:=j-1;
        end;
    end.

```

b. Eficiența ca timp de execuție se obține aplicând algoritmul clasic de interclasare a 2 vectori, singura modificare fiind aceea că vectorii sunt parcurși din “direcții” diferite (unul de la poziția 1 iar celălalt de la poziția k2). Pentru a avea și eficiență din punct de vedere al spațiului de memorie utilizat, am memorat în cei doi vectori doar elementele pare; primul tablou având k1 numere pare iar al doilea k2.

Varianta <98 neintensiv>

SUBIECTUL III

4.

a)

Varianta C/C++

```

#include<fstream.h>
ifstream f("date.txt");
int a[10];

void m(x)
{
    while(f>>x)
        a[x]++;
for(i=0;i<=9;i++)
    for(k=1;k<=a[i];k++)
        cout<<i<<" ";
}

```

Varianta PASCAL

```

var a:array[0..9] of integer;
    i,k,x:integer;
    f:text;

begin
    while(not eof(f)) do
        begin
            read(f,x);
            Inc(a[x]);
        end;
    for i:=0 to 9 do
        for k:=1 to a[i] do
            write(i, ' ');
        end.

```

b. Eficiența ca timp de execuție este dată de faptul că nu este utilizat nici un algoritm de sortare și mai mult, nici nu sunt memorate elementele citite. Astfel este utilizat un vector cu 10 componente, fiecare element valori posibile din șirul dat. Pentru fiecare element nou vom incrementa pe poziția egală cu elementul citit. Urmează să afișăm șirul componentelor aflate pe pozițiile 0,1,...,9 de un număr de ori egal cu a[0],a[1],...,a[9].

SUBIECTUL I

1. d)

2.

b) x poate fi 1, 3, 5, 7, 9

c)

```

citește x ; (x număr natural)
p=(4^x)%10;
scrie p;
    
```

Obs: Prin 4^x am notat 4 la puterea x.

Pentru x suficient de mare (natural) algoritmul nu oferă rezultatul corect (depășire de tip).
Este recomandată varianta în care sunt calculate puterile lui 4 în funcție de forma

4^{2k} are ca ultimă cifră 6, iar 4^{2k+1} are ultima cifră 4.

Așadar, varianta corectă și eficientă este următoarea :

```

Citeste x ; (x numar natural)
daca x numar par
    Scrie 6
altfel
    Scrie 4
    
```

d)

Varianta C/C++

```

#include<fstream.h>
void main()
{unsigned long x;
cin>>x;
if(x%2==0) cout<<6;
else cout<<4;
}
    
```

Varianta Pascal

```

var x:longint;
begin
readln(x);
if(x mod 2=0) then
writeln(6)
else
writeln(4);
end.
    
```

SUBIECTUL II

1. d)

2. a)

3. Considerăm graful format din muchiile (1,2),(1,3),(4,5). Vârfurile 2,3,4,5 au gradul 1.

Matricea de adiacență este :

```

0 1 1 0 0
1 0 0 0 0
1 0 0 0 0
0 0 0 0 1
0 0 0 1 0
    
```

4.

	1	3	5	
EL	3	5		
AD 4	3	5	4	
AD 6	3	5	4	6

5. Explicații

Vom parcurge doar yona de sub diagonala principalăa matricei. Fiecare element va fi egal cu numărul de ordine al liniei pe care se află. ($a[i][j]=i$) Elementele de deasupra diagonalei principale vor fi egale cu numărul de ordine al coloanei pe care se află($a[j][i]=i$).

Varianta C/C++

```
#include<fstream.h>
int a[51][51],n,i,j;

void main()
{ do{cin>>n;}while(n<=0 || n>50);

for(i=1;i<=n;i++)
for(j=1;j<=i;j++)
a[i][j]=a[j][i]=i;

for(i=1;i<=n;i++)
{ for(j=1;j<=n;j++)cout<<a[i][j]<<" ";
cout<<endl;
}
}
```

Varianta PASCAL

```
var a:array [1..51,1..51]of integer;
n,i,j:integer;
begin
repeat
readln(n);
until(n>0) and (n<=50);

for i:=1 to n do
for j:=1 to i do
begin
a[i][j]:=i;
a[j][i]:=i;
end;
for i:=1 to n do
begin
for j:=1 to n do
write(a[i][j],' ');
writeln;
end;
end.
```

SUBIECTUL III

1. a)
2. wt și zy

3. Explicații

Numărul k este convertit în șir de caractere pentru a permite eliminarea simplăa primei , respectiv ultimei cifre din șir. Condiția ca cifrele numărului să fie nenule este astfel inutilă. Șirul obținut după eliminare este convertit în număr.

Varianta C/C++

```
#include<fstream.h>
#include<stdlib.h>
#include<string.h>
char s[10];
void nr(long &k)
{ ltoa(k,s,10);

strcpy(s,s+1);
s[strlen(s)-1]='\0';
k=atol(s);
}
void main()
{ long n=12438;
nr(n);
cout<<n;
}
```

Varianta PASCAL

```
var s:string[10];
n:longint;
procedure nr(var k:longint);
var l,cod:integer;
begin
str(k,s);

l:=length(s);
delete(s,l,1);
delete(s,1,1);
val(s,k,cod);
end;

begin
n:=1243978;
nr(n);
writeln(n);
end.
```

4.

a) Explicații

Problema constă în determinarea subsecvenței maxime de elemente cu o anumită proprietate (în cazul nostru alternarea parității).

Varianta C/C++

```
#include<fstream.h>
long a[3001],n,i,pmax,lmax,l,p,x;
ifstream f("date.txt");

void main ()
{ while(f>>x)
  {n++;
   a[n]=x;
  }

  pmax=1;
  lmax=0;

  for(int i=1;i<=n;i++)
  {p=i;
   l=1;
   while((a[i]+a[i+1])%2==1 &&
(i<=n-1))
   { l++;
    l++;
   }
   if (l>lmax)
   {pmax=p;
    lmax=l;
   }
  }
  cout<<endl;
  for(i=pmax;i<=lmax+pmax-1;i++)
  cout<<a[i]<<" ";
}
```

Varianta PASCAL

```
var f:text;
    a:array[1..3001] of longint;
    n,i,pmax,lmax,l,p,x:integer;

begin
  assign(f,'date.txt');
  while(not eof(f)) do
  begin
    read(f,x);
    n:=n+1;
    a[n]:=x;
  end;

  pmax:=1;
  lmax:=0;

  i:=1;

  while (i<=n) do
  begin
    p:=i;
    l:=1;

    while((a[i]+a[i+1]) mod 2=1) and
(i<=n-1) do
    begin
      i:=i+1;
      l:=l+1;
    end;

    if (l>lmax) then
    begin
      pmax:=p;
      lmax:=l;
    end;
    inc(i);
  end;
  writeln;
  for i:=pmax to lmax+pmax-1 do
  write(a[i],' ');
end.
```

b.)

Eficiența din punct de vedere a timpului de execuție, constă într-o singură parcurgere a elementelor tabloului, adică un algoritm de complexitate $O(n)$. Pentru fiecare element avem o secvență de lungime cel puțin 1 care începe de la poziția respectivă. Cât timp elementul următor are paritate diferită față de anteriorul său, mărim lungimea subsecvenței și trecem la elementul următor.

În cazul în care găsim 2 elemente vecine cu aceeași paritate, secvența nu mai poate fi mărită; vom compara lungimea secvenței curente cu lungimea maximă determinată până în acel moment și eventual actualizăm lungimea maximă și poziția acesteia $pmax$. Parcurgerea continuă cu elementul următor care face parte dintr-o nouă subsecvență de lungime cel puțin 1. După parcurgerea tuturor elementelor vectorului vom avea în variabila $lmax$ și $pmax$ lungimea respectiv poziția de început a subsecvenței cerute (soluția problemei).

SUBIECTUL II

1. b)

SUBIECTUL III

a.) Explicații

Problema constă în determinarea subsecvenței maxime de elemente cu o anumită proprietate (în cazul nostru alternarea parității).

Varianta C/C++

```
#include<fstream.h>
ifstream f("date.txt");
long a[1001],x;
int i,j,n,k,poz[1001];
void sortare(long v[1001], int n)
{ for(int i=1;i<=n-1;i++)
  {long min=v[i];
  int poz=i;
  for(int j=i+1;j<=n;j++)
    if(v[j]<min)
      { min=v[j];
      poz=j;
      }
  v[poz]=v[i];
  v[i]=min;
  }
}
void afis_0(int p)
{ for(int i=1;i<=p;i++)
  cout<<0<<" ";
}
void main()
{
  f>>n;
  for(i=1;i<=n;i++)
    {f>>x;
    if(x>0)
      {k++;
      a[k]=x;
      poz[k]=i;
      }
  }
  sortare(a,k);
  for(i=1;i<=k;i++)
    {afis_0(poz[i]-poz[i-1]-1);
    cout<<a[i]<<" ";
    }
  afis_0(n-poz[k]);
}
```

Varianta PASCAL

```
type vector=array[1..1001] of
longint;
var f:text;
    x:longint;
    a:vector;
    i,j,n,k:integer;
    poz:array[1..1001]of integer;

procedure sortare(v:vector;
n:integer);
var i,j,poz:integer;
    min:longint;
begin
  for i:=1 to n-1 do
    begin
      min:=v[i];
      poz:=i;
      for j:=i+1 to n do
        if(v[j]<min) then
          begin
            min:=v[j];
            poz:=j;
          end;
      v[poz]:=v[i];
      v[i]:=min;
    end;
end;

procedure afis_0(p:integer);
var i:integer;
begin
  for i:=1 to p do
    write(0,' ');
end;

begin
  assign(f,'date.txt');
  reset(f);
  readln(f,n);
  for i:=1 to n do
    begin
      read(f,x);
      if(x>0) then
        begin
          k:=k+1;
          a[k]:=x;
          poz[k]:=i;
        end;
    end;
end;
```

```

        end;
    end;
    sortare(a, k);

    for i:=1 to k do
    begin
        afis_0(poz[i]-poz[i-1]-1);
        write(a[i], ' ');
    end;
    afis_0(n-poz[k]);
end.
```

b.)

Eficiența din punct de vedere a spațiului de memorie se datorează faptului că nu vor fi memorate decât valorile ce urmează să fie ordonate, adică acele strict pozitive. Simultan cu memorarea acestor valori vom memora și poziția acestuia în fișier, adică numărul de ordine din fișier. Afișarea valorilor 0 în locul celor negative se deduce pe baza pozițiilor a două numere pozitive alăturate din șirul ordonat crescător.

Varianta <100>

SUBIECTUL I

1. a.)
- 2.
- a) 1
- b) 106, 997
- c)

Varianta C/C++

```

#include<fstream.h>
void main()
{long a;
 int b;
 cin>>a;
 do
 {b=0;
 while(a)
 {b=b+a%10;
 a=a/10;
 }
 a=b;
 }while(a>=10);
 cout<<b;
 }
```

Varianta Pascal

```

var f:text;
    a:longint;
    b:integer;
begin
    readln(a);
    repeat
        b:=0;
        while(a<>0) do
            begin
                b:=b+a mod 10;
                a:=a div 10;
            end;
        a:=b;
    until(a<10);
    writeln(b);
end.
```

d) citește a ; (a număr natural mai mic ca 10^9)

```

b←0;
cat timp (a>0 sau b>=10)
{ b←b+a%10;
  a←[a/10];
  daca(b>=10)

      daca(b=0)
      b=0;
```

```

    }
};
scrie b;

```

SUBIECTUL II

1. d)
2. a)
3. 2, 4, 6
- 4.

Varianta C/C++

Strchr(e, '+') reprezinta sirul "+73"
 Strchr(e, '+')+2 reprezinta sirul "3" care este copiat peste e.

5. Explicații

Vom testa pentru fiecare element din matrice dac to i vecinii acestuia sunt strict mai mici decât el. (doar vecinii care fac parte din matrice). ă ț

Varianta C/C++

```

#include<fstream.h>
#include<string.h>
#include<stdlib.h>
ifstream f("mat.in");
long a[101][101];
int n,i,j;
int verifica(int i,int j)
{ //verifica daca elementul
a[i][j]
// este strict mai mic decat
vecinii lui
long x=a[i][j];
if((i+1<=n)&&
(x>=a[i+1][j]))return 0;
if((i-1>=1)&& (x>=a[i-
1][j]))return 0;
if((j-1>=1)&& (x>=a[i][j-
1]))return 0;
if((j+1<=n)&&
(x>=a[i][j+1]))return 0;

return 1;

}
void main()
{ do{cin>>n;}while(n<=1 ||n>=100);
for(int i=1;i<=n;i++)
for(int j=1;j<=n;j++)
cin>>a[i][j];

for(i=1;i<=n;i++)
for(j=1;j<=n;j++)
if (verifica(i,j)==1)
cout<<a[i][j]<<" ";
}

```

Varianta PASCAL

e='3'

Varianta PASCAL

```

var a:array[1..101,1..101] of longint;
i,j,n:integer;

function
verifica(i:integer;j:integer):integer;
var x:longint;
rez:integer;
begin
{verifica daca elementul a[i][j]
este strict mai mic decat vecinii
lui}
rez:=1;
if((i+1<=n) and (x>=a[i+1][j])) then
rez:=0;
if((i-1>=1) and (x>=a[i-1][j])) then
rez:=0;
if((j-1>=1) and (x>=a[i][j-1])) then
rez:=0;
if((j+1<=n) and (x>=a[i][j+1])) then
rez:=0;

verifica:=rez;

end;

begin
repeat
readln(n);
until((n>1) and (n<100));

for i:=1 to n do
for j:=1 to n do
read(a[i][j]);

for i:=1 to n do
for j:=1 to n do
if (verifica(i,j)=1) then
write(a[i][j], ' ');

writeln;

end.

```

SUBIECTUL III

1. b)
2. 531024
- 3.

Varianta C/C++

```
void schimb(long &x,long &y)
{long aux=x;x=y;y=aux;}

void ordonare(long &x,long &y,long
&z)
{ if(x>y) schimb(x,y);
  if(x>z) schimb(x,z);
  if(y>z) schimb(y,z);
}

int divizor(long a,long b,long c)
{ ordonare(a,b,c);

  int nr=0;
  if (b%a+c%a ==0) nr=1;
  for(long d=a/2;d>=1;d--)
    if(a%d + b%d + c%d ==0)
      {nr++;
       cout<<d<<" ";
      }

  return nr;
}
```

4. Explicații

a.)

Varianta C/C++

```
#include <fstream.h>
#include <stdlib.h>
#include <string.h>

ifstream f("sir.txt");
char
s1[5001],s2[5001],k[5001],s[5001];
int x,p;
void citire()
{f>>x>>p;
 f.get();
 f.get(k,5001);
 f.close();
}

void rezolvare()
```

Varianta PASCAL

```
procedure schimb(var
x:longint;y:longint);
var aux:longint;
begin
  aux:=x;
  x:=y;
  y:=aux;
end;

procedure ordonare( var
x,y,z:longint);
begin
  if(x>y) then schimb(x,y);
  if(x>z) then schimb(x,z);
  if(y>z) then schimb(y,z);
end;

function divizor(
a,b,c:longint):integer;
var nr:integer;
d:longint;
begin
  ordonare(a,b,c);
  nr:=0;
  if (b mod a+c mod a)=0 then
    nr:=1;
  for d:=a div 2 downto 1 do
    if(a mod d + b mod d + c mod d
=0) then
      begin
        nr:=nr+1;
        write(d, ' ');
      end;
  divizor:=nr;
end;
```

Varianta PASCAL

```
var s1,s2,s:string[255];
x,k:word;

procedure rezolvare();

begin
  str(x,s1);
  str(x+1,s2);

  while(sir<>s2) do
    begin
      s:=s2+s1;
      s1:=copy(s2,1,length(s2));
      s2:=copy(s,1,length(s));
    end;
```

<pre> { ltoa(x, s1, 10); ltoa(x+1, s2, 10); while(strcmp(k, s2)) { strcpy(s, s2); strcat(s, s1); strcpy(s1, s2); strcpy(s2, s); } cout<<s1<<endl; } void main() { citire(); rezolvare(); } </pre>	<pre> writeln(s1); end; begin assign(f, 'sir.txt'); reset(f); readln(f, x, k); readln(f, sir); rezolvare(); end. </pre>
--	--

b.) În variabilele s1 și s2 am memorat primii 2 termeni ai șirului ce urmează să fi construit. Următorul termen al șirului se obține prin concatenarea șirului s1 la s2. De aceea au fost folosite șirurile de caractere. Fiecare nou termen al șirului este testat cu șirul căutat. Eficiența din punct de vedere a memoriei utilizate constă în faptul că nu e nevoie memorarea tuturor șirurilor construite, ci doar a ultimilor două pe baza cărora se poate construi un termen nou.

Eficiența din punct de vedere a timpului de execuție provine din faptul că vor fi construite elementele șirului o singură dată. (Problema este asemănătoare cu determinarea celui de-al n-lea termen din șirul lui fibonacci). Dacă termenul căutat este găsit, atunci nu mai este necesară construirea celorlalte elemente (enunțul problemei asigură existența soluției). Complexitatea este $O(n)$, unde n este numărul de ordine, în șirul definit în enunț, al termenului care trebuie determinat.

Varianta <100 neintensiv>

SUBIECTUL II

1. d)
2. a)
- 4.

Varianta C/C++

`e+strlen(e)-1` reprezintă șirul "3" care este copiat peste e.

Varianta PASCAL

'3'

SUBIECTUL III

4. Explicații

a.)

Varianta C/C++

```

#include<fstream.h>
ifstream f("bac.txt");
long x, a, b;
int poza, pozb, nr;
void main()
{ do
  { f>>x; nr++;
  } while(x%2==0);

  a=x; poza=nr;
  do
  { f>>x; nr++;
  } while(x%2==0);

```

Varianta PASCAL

```

var f:text;
    x, a, b:longint;
    poza, pozb, nr:integer;
begin
  assign(f, 'bac.txt');
  reset(f);
  repeat
    read(f, x);
  until (x mod 2=1);
  a:=x;
  poza:=nr;

```

```

b=x;pozb=nr;

while(f>>x)
{ nr++;
  if(x%2==1)
  {a=b;
    b=x;
    poza=pozb;
    pozb=nr;
  }
}
cout<<a<<" are numar de
ordine"<<poza;
}

```

```

repeat
  read(f,x);
  nr:=nr+1;
until(x mod 2=1);
b:=x;
pozb:=nr;

while(not eof(f)) do
begin
  read(f,x);
  nr:=nr+1;
  if(x mod 2=1) then
  begin
    a:=b;
    b:=x;
    poza:=pozb;
    pozb:=nr;
  end;
end;

write(a,' are numar de ordine
',poza);
end.

```

b.) Vom citi din fișier numerele fără le memora și vom determina primele 2 numere impare și poziția lor în cadrul șirului. (enunțul asigură existența a cel puțin 2 astfel de numere). Pentru fiecare număr impar nou vom actualiza valorile și pozițiile ultimilor 2 numere impare.

Algoritmul este eficient din punct de vedere a memoriei deoarece nu utilizează decât câteva variabile (nu se folosesc tablouri pentru stocarea elementelor).

Algoritmul este liniar, adică dintr-o singură parcurgere se obțin rezultatele dorite. ($O(n)$)

unde n este numărul elementelor din fișier.

