 **!!! ATENȚIE !!!** 

Aceste rezolvări NU au fost aprobate de MINISTERUL EDUCAȚIEI sau altă comisie recunoscută de Ministerul Educației. În consecință nimeni nu își asumă răspunderea pentru eventualele greşeli şi / sau perderi survenite în urma folosirii lor!

Foloseşte rezolvările pe riscul tău !!!

Dacă găseşti greşeli sau ai nelămuriri în legătură cu o anumită rezolvare trimite-mi un e-mail pe adresa raducu@trei.ro şi voi încerca să lămuresc / corectez problema.

Varianta 1:

1. b **2.** 13245

3.
```
var n,x:integer;
    f:text;
    gasit:boolean;
begin
  gasit:=false;
  assign(f,'bac.txt'); reset(f);
  write(' n= '); read(n);
  while not eof(f) do
    begin
      read(f,x);
      if x mod n = 0
         then  begin
                 write(x,' ');
                 gasit:=true;
               end;
    end;
  close(f);
  if gasit = false
    then write(' NU EXISTA ! ');
end.
```

4.
```
type sir=array[1..100]of integer;
var a:sir;
    n,i:integer;
    gasit:boolean;
function sub(v:sir;n,a:integer):integer;
var e,i:integer;
begin
  e:=0;
  for i:=1 to n do
    if v[i]<a
      then e:=e+1;
  sub:=e;
end;
begin
  write(' N= '); read(n);
  for i:=1 to n do
    begin
      write(' A[',i,']= ');
      read(a[i]);
    end;
  repeat
    ok:=true;
    for i:=1 to n do
      if v[i]>v[i+1]
        then begin
```

```
                ok:=false;
                v[i]:=v[i]+v[i+1];
                v[i+1]:=v[i]-v[i+1];
                v[i]:=v[i]-v[i+1];
              end;
  until ok;
ok:=true;
for i:=2 to n do
  if sub(v,n,v[i])<>i-1
    then ok:=false;
if ok
  then write(' DA')
  else write(' NU');
end.
```

Varianta 2:

1. d **2.** 5310153

3.

```
var a:array[1..100]of integer;
    f:text;   sortat:boolean;
    n, nr, i, aux:integer;
begin
  assign(f,'nr.txt');   reset(f);
  n:=0;
  while not eof(f) do
    begin
      read(f,nr);
      if nr >0
        then begin
               n:=n+1;
               a[n]:=nr;
             end;
    end;
  close(f);
  if n=0 then write(' NU EXISTA')
         else begin
                repeat
                  sortat:=true;
                  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;
                             sortat:=false;
                           end;
                until sortat;
                for i:=1 to n do
```

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

4.
```
  var n,n1:integer;
  function f(a:integer):integer;
  var i,s,d:integer;
  begin
    d:=2; s:=0;
    while a>1 do
      begin
        while a mod d=0 do
          begin
            a:=a div d;
            s:=s+1;
          end;
        d:=d+1;
      end;
    f:=s;
  end;
  begin
    write(' n= '); read(n);
    if n<10
      then begin n1:=2; n2:=2; end
      else if n<100
             then begin
                    n1:=n mod 10 *10 + n div 10;
                    n2:=2;
                  end
             else begin
                    n1:=n mod 10 *10 + n div 10;
                    n2:=n1 mod 10 *10 + n1 div 10;
                  end;
    if (f(n)=1) and (f(n1)=1) and (f(n2)=1)
      then write(' DA ')
      else write(' NU ');
  end.
```

Varianta 3:

1. a **2.** xyyy

3.
```
  var a:array[1..100]of integer;
      f:text;   sortat:boolean;
      n, nr, i, aux:integer;
  begin
    assign(f,'nr.txt');   reset(f);
    n:=0;
    while not eof(f) do
```

```
      begin
        read(f,nr);
        if (nr > 99) or (nr < -99)
          then begin
                 n:=n+1;
                 a[n]:=nr;
               end;
      end;
    close(f);
    if n=0 then write(' NU EXISTA')
           else begin
                  repeat
                    sortat:=true;
                    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;
                               sortat:=false;
                             end;
                  until sortat;
                  for i:=1 to n do
                    write(a[i],' ');
                end;
end.
```

4.

```
var n ,c, i,n1,n2:longint;
function cif(a:longint; b:byte):byte;
var na:byte;
begin
  na:=0;
  while a>0 do
    begin
      if a mod 10 = b
        then na:=na+1;
      a:=a div 10;
    end;
  cif:=na;
end;
begin
  write(' n= '); read(n);
  n1:=0;
  for i:=9 downto 0 do
  begin
    c:=cif(n,i);
    if (c>0) and (c mod 2<>0)
      then begin
             n1:=0; exit;
           end
```

```
        else if c>0 then begin
                           if c=2
                             then n1:=n1*10+i;
                           if c=4
                             then n1:=n1*100+i*11;
                           if c=6
                             then n1:=n1*1000+i*111;
                           if c=8
                             then n1:=n1*10000+i*1111;
                         end;
    end;
    n2:=n1;
    while n1>0 do
      begin
        n2:=n2*10+n1 mod 10;
        n1:=n1 div 10;
      end;
    writeln(n2);
  end.
```

Varianta 4:

1. c **2.** 01111

3.

```
var a:array[1..100]of integer;
    f:text;   sortat:boolean;
    n, nr, i, aux:integer;
begin
  assign(f,'nr.txt');   reset(f);
  n:=0;
  while not eof(f) do
    begin
      read(f,nr);
      if (nr < 100)
        then begin
               n:=n+1;
               a[n]:=nr;
             end;
    end;
  close(f);
  if n=0 then write(' NU EXISTA')
         else begin
                repeat
                  sortat:=true;
                  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;
```

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

4.

```
var n ,c, i,n1,n2:longint;
function cif(a:longint; b:byte):byte;
var na:byte;
begin
  na:=0;
  while a>0 do
    begin
      if a mod 10 = b
        then na:=na+1;
      a:=a div 10;
    end;
  cif:=na;
end;
begin
  write(' n= '); read(n);
  n1:=0;
  for i:=1 to 9 do
  begin
    c:=cif(n,i);
    if (c>0) and (c mod 2<>0)
      then begin
             n1:=0; exit;
           end
      else if c>0 then begin
                         if c=2
                           then n1:=n1*10+i;
                         if c=4
                           then n1:=n1*100+i*11;
                         if c=6
                           then n1:=n1*1000+i*111;
                         if c=8
                           then n1:=n1*10000+i*1111;
                       end;
  end;
  n2:=n1;
  while n1>0 do
    begin
      n2:=n2*10+n1 mod 10;
      n1:=n1 div 10;
    end;
  writeln(n2);
end.
```

Varianta 5:

1. d **2.** 7*****

3.
```
var n:longint;
    f:text;
begin
  assign(f,'nr.txt'); rewrite(f);
  write(' n= '); read(n);
  while n>0 do
    begin
      write(f,n,' ');
      n:=n div 10;
    end;
  close(f);
end.
```

4.
```
var n,i:integer;  gasit:boolean;
    a:array[1..100]of longint;
function f(a:longint):longint;
var d:longint;
begin
  d:=2;
  while a mod d <>0 do d:=d+1;
  f:=d
end;
begin
  write(' n= '); read(n);
  for i:=1 to n do
    begin
      write(' A[',i,']= ');
      read(a[i]);
    end;
  gasit:=false;
  for i:=1 to n do
    if f(a[i]) = a[i]
      then begin
             gasit:=true;
             write(a[i],' ');
           end;
  if not gasit
    then write(' NU EXISTA! ');
end.
```

Varianta 6:

1. b **2.** 7

3.
```
var a:array[1..100]of integer;
    i,n,s:integer;
begin
  write(' n= '); read(n);
  for i:=1 to n do
    begin
      write(' A[',i,']= ');
      read(a[i]);
    end;
  s:=0;
  for i:=1 to n do s:=s+a[i];
  for i:=n downto 1 do begin
                         writeln(s);
                         s:=s-a[i];
                       end;
end.
```

4.
```
var f:text;
    n1,n2,na:longint;
begin
  assign(f,'bac.txt'); reset(f);
  read(f,n1); na:=1;
  while not eof(f) do
    begin
      read(f,n2);
      if n1<>n2
        then begin
               write(n1,' ',na,' ');
               n1:=n2; na:=1;
             end
        else na:=na+1;
    end;
  write(n1,' ',na);
  close(f);
end.
```

Varianta 7:

1. c **2.** 126

3.
```
var a:array[1..300] of integer;
    n, i, aux, ii, ip:integer;
begin
  write(' n= '); read(n);
  for i:=1 to 3*n do
    begin
      write(' A[',i,']= ');
      read(a[i]);
    end;
```

```
    ip:=0; ii:=0;
    for i:=1 to n do
      if ( a[i] mod 2 = 0 ) and ( ip = 0 )
        then ip:=i;
    for i:=3*n downto 2*n+1 do
      if ( a[i] mod 2 = 1 ) and ( ii = 0 )
        then ii:=i;
    aux:=a[ii];
    a[ii]:=a[ip];
    a[ip]:=aux;
    for i:=1 to 3*n do
      write(a[i],' ');
  end.
```

4.
```
var s,n:longint;
    g:text;

function sub(n:longint):longint;
var x,y:longint;
begin
  x:=1;
  while x<n do
    begin
      if x<5
        then x:=x+1
        else x:=2*x;
    end;
  if n>5
    then f:=x div 2
    else f:=x;
end;

begin
  write(' s= '); read(s);
  while s>1 do
    begin
      n:=f(s);
      write(n,' ');
      s:=s-n;
    end;
end.
```

Varianta 8:

1. b

2.
```
2
1 2 3
```

3.
```
function sub(n:byte):integer;
var s,i,x:integer;
begin
  s:=0;
```

```
    for i:=1 to n do
      begin
        read(x);
        if sqrt(x)=trunc(sqrt(x))
          then s:=s+x;
    sub:=s;
  end;
```

4.a.

```
var a,b:array[1..100]of integer;
    i,j,n,m,ua:integer;
    f:text;

begin
  assign(f,'bac.txt'); rewrite(f);
  write(' n= '); read(n);
  for i:=1 to n do
    begin
      write(' A[',i,']= ');
      read(a[i]);
    end;
  write(' m= '); read(m);
  for j:=1 to m do
    begin
      write(' B[',j,']= ');
      read(b[j]);
    end;
  i:=1;  j:=1;   ua:=-a[1]-b[1];
  while (i<=n) and (j<=m) do
    begin
      if a[i]<b[j]
        then begin
               if ua mod 2<>a[i] mod 2
                 then begin
                        write(f,a[i],' ');
                        ua:=a[i];
                      end;
               i:=i+1;
             end
        else begin
               if ua mod 2<>b[j] mod 2
                 then begin
                        write(f,b[j],' ');
                        ua:=b[j];
                      end;
               j:=j+1;
             end;
    end;
  while i<=n do
    begin
      if ua mod 2<>a[i] mod 2
        then begin
               write(f,a[i],' ');
```

```
                  ua:=a[i];
                end;
          i:=i+1;
        end;
      while j<=m do
        begin
          if ua mod 2<>b[j] mod 2
            then begin
                   write(f,b[j],' ');
                   ua:=b[j];
                 end;
          j:=j+1;
        end;
      close(f);
    end.
```

4.b. Programul ales se bazează pe interclasarea a două siruri ordonate anterior. La fiecare pas va copia in fisier cea mai mica valoare dintre a[i] şi b[j] numai daca difere de paritatea celei anterioare, avansând cu o poziție într-unul din şiruri.

Programul este eficient din punc de vedere al timpului de execuție deoarece citeşte o singură dată numerele, nu apelează la metode de sortare ci scrie direct, parcurgând o singura data fiecare sir, valorile cerute.

Varianta 9:

1. b **2.** 1

3.
```
procedure sub(n:integer;k:integer);
var i:integer;
begin
  for i:=n downto 1 do
    write(n*k);
end;
```

4.a.
```
var f:text;
    a:array[1..1000]of boolean;
    x,i:longint;

begin
  assign(f,'bac.txt'); reset(f);
  while not eof(f) do
    begin
      read(f,x);
      if x<1000
        then a[x]:=true;
    end;
  i:=999;
  while a[i] and (i>0) do i:=i-1;
  write(i,' '); i:=i-1;
```

```
  while a[i] and (i>0) do i:=i-1;
  write(i,' ');
  close(f);
end.
```

4.b. Folosesc un sir pentru a reține numerele mai mci de 1000 care apar in fisier apoi parcurgand sirul de la sfarsit spre inceput afişez primele două valori care nu s-au gasit in fisier (au in sir false)

Programul este eficiend din punct de vedere al timpului de executare deoarece citeşte numai o dată datele din fişier.

Varianta 10:

1. a **2.** 5

3.
```
procedure sub(n:integer; var a,b:integer);
var prim:boolean;
    d:integer;
begin
  a:=0;  b:=0;
  while (a=0) or (b=0) do
    begin
      prim:=true;
      for d:=2 to n div 2 do
        if n mod d = 0
          then prim:=false;
      if prim
        then if a=0
               then a:=n
               else b:=n;
      n:=n-1;
    end;
end;
```

4.a)
```
var p:array[1..9999]of integer;
    f:text;
    t,v,c:integer;

begin
  assign(f,'produse.txt'); reset(f);
  while not eof(f) do
    begin
      read(f,t,v,c);
      p[t]:=p[t]+v*c;
    end;
  close(f);
  for t:=1 to 9999 do
    if p[t]>0
      then writeln(t,' ',p[t]);
end.
```

4.b) Pentru fiecare produs t, în şirul p pe poziția t voi reține produsul dintre cantitatea şi prețul unui produs. La final afişez numai tipurile de produse care au valoarea p[t]>0

Programul este eficient deoarece nu foloseşte structuri de date pentru a reține toate elementele şirului (toate datele despre produse) şi fişierul de intrare este citit o singură data. În concluzie programul consumă puțină memorie şi este rapid.

Varianta 11:

1. b **2.** 1

3.a. Citim pe rând câte un număr din fişier. Dacă numărul nou citi este mai mare decât maximul de pâna atunci il reținem ca fiind maxim. După fiecare citire afişăm maximul.

Programul este eficient din punct de vedere al timpului de execuție deoarece citeste o singura data numerele din fisier şi nu reține toate numerele deci este eficient si din punct de vedere al spațiului de memorie utilizat.

3.b.
```
var f:text;
    n,i,x,xm:longint;
begin
  xm:=0;
  assign(f,'numere.txt'); reset(f);
  readln(f,n);
  for i:=1 to n do begin
                     read(f,x);
                     if xm<x
                        then xm:=x;
                     write(xm,' ');
                   end;
  close(f);
end.
```

4.
```
var i,n,c,x:longint;
function sum(x:longint):integer;
var s,d:longint;
begin
  s:=1;
  for d:=2 to x div 2 do
    if x mod d = 0
      then s:=s+d;
  sum:=s+x;
end;
begin
  write(' n= '); read(n);
  c:=0;
```

```
    for i:=1 to n do
      begin
        write(' nr= '); read(x);
        if (sum(x)=x+1) and (x>1)
          then c:=c+1;
      end;
    write(' c= ',c);
  end.
```

Varianta 12:

1. c **2.** 2+2+2+3; 2+2+5; 2+7

3.a) Mă bazez pe algoritmul interclasarii a două şiruri. Pâna când nu am terminat de citi valorile din ambele fisiere citesc noi valor apoi afisez valoarea mai mica dintre cele două si divizibilă cu 5.

Programul este eficient deoarece nu reține toate elementele celor două fişiere ci numai câte o valoare, pe rând, deci consumă puțnnă memorie şi percurge fişierele o singură data deci este rapid.

3.b)

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

```
        then write(x);
      close(f);  close(g);
    end.
```

4.
```
var n,i,ne,x:longint;
    ok:boolean;
begin
  write(' n= '); read(n);
  ne:=0;
  for i:=1 to n do
    begin
      read(x);
      ok:=true;
      while x>9 do
        begin
          if x mod 10 <> x div 10 mod 10
             then ok:=false;
          x:=x div 10;
        end;
      if ok
        then ne:=ne+1;
    end;
  write(' Numere cu toate cifrele egale: ',ne);
end.
```

Varianta 13:

1. d **2.** 6

3.
```
var p,mg,n:integer;
begin
  write(' n= '); read(n);
  mg:=0; p:=0;
  while p<n do
    begin
      mg:=mg+1;
      p:=p+mg;
    end;
  write(p-n+1);
end.
```

4.
```
var f,g:text;
    nr:longint;
procedure P(var n:longint; c:byte);
var i:integer;
    s,s2:string[10];
begin
  str(n,s); str(c,s2);
  for i:=1 to length(s) do
```

```
      if s[i]=s2
        then delete(s,i,1);
    val(s,n,i);
  end;

  begin
    assign(f,'bac.in'); reset(f);
    assign(g,'bac.out'); rewrite(g);
    while not eof(f) do
      begin
        read(f,nr);
        P(nr,1);
        P(nr,3);
        P(nr,5);
        P(nr,7);
        P(nr,9);
        if nr>0 then write(g,nr,' ');
      end;
    close(f);
    close(g);
  end.
```

Varianta 14:

1. b **2.** 72

3.
```
var a:array[0..9]of integer;
    c,i,j:byte;
    n:integer;

begin
  write(' n= '); read(n);
  write(' Introduceti cifrele:   ');
  for i:=1 to n do begin
                     read(c);
                     a[c]:=a[c]+1;
                   end;
  for i:=0 to 9 do
    for j:=1 to a[i] do
      write(i,' ');
end.
```

4.
```
var c,sp,x,s:integer;
    f:text;

begin
  assign(f,'bac.txt');   reset(f);
  c:=0; sp:=0;
  while not eof(f) do
    begin
      read(f,x);
```

```
        write(x,' ');c:=c+1;
        if c mod 5 = 0
          then writeln;
        s:=0;
        while x<>0 do begin
                        s:=s+x mod 10;
                        x:=x div 10;
                      end;
        if s mod 2 = 0
          then sp:=sp+1;
      end;
    writeln; write(sp);
    close(f);
  end.
```

Varianta 15:

1. b **2.** 85

3.
```
var nr_div_max, nr_div, i,j,nd,n:integer;
begin
  write(' n= '); read(n);
  nr_div_max:=0; nr_div:=0;
  for i:=1 to n do
    begin
      nd:=0;
      for j:=1 to i do
        if i mod j = 0
          then nd:=nd+1;
      if nd>nr_div
        then begin
               nr_div:=nd;
               nr_div_max:=i;
             end;
    end;
  write(nr_div_max);
end.
```

4.a. Pornesc la citirea fisierului. Pentru fiecare umar citit verific daca este prim sau nu. In cazul in care am mai descoperit un numar prim salvez numarul prim anterior descoperit in.variabila pp si noul numar prim in variabila up. La final verific daca am gasit cel putin doua numere prime si afisez datele in consecință.

Programul este eficient din punct de vedere al timpului de executie deoarece numerele sunt citite si imediat prelucrate ne fiind necesare alte citiri ulterioare.

Programul este eficient din punct de vedere al memoriei consumate deoarece nu foloseste structuri pentru a reține numerele.

4.b.
```
var pp,up,i,x:integer;
    prim:boolean;
```

```
      f:text;
  begin
    assign(f,'bac.in');   reset(f);
    up:=0;
    while not eof(f) do
      begin
        read(f,x);
        prim:=true;
        for i:=2 to x div 2 do
          if x mod i = 0
            then prim:=false;
        if prim then begin
                       pp:=up;
                       up:=x;
                     end;
      end;
    close(f);
    if pp>0
      then write(pp,' ',up)
      else write('Numere prime insuficiente');
  end.
```

Varianta 16:

1. d **2.** 77755, 77757, 77777

3.
```
type sir=array[1..100]of integer;

function multiplu(a:sir; n,k:integer):integer;
var c,i:integer;
begin
  c:=0;
  for i:=1 to n do
    if (a[i] mod k=0) and (a[i] mod 10=k)
      then c:=c+1;
  multiplu:=c;
end;
```

4.
```
var a:array[0..9]of integer;
    f:text;
    n:longint;
    c,i,j:integer;
begin
  assign(f,'numere.txt'); reset(f);
  while not eof(f) do
    begin
      read(f,n);
      while n>0 do begin
                     c:=n mod 10;
```

```
                        a[c]:=a[c]+1;
                        n:=n div 10;
                      end;
      end;
    for i:=9 downto 0 do
      for j:=1 to a[i] do
        write(i);
    close(f);
  end.
```

Varianta 17:

1. c **2.** 12347, 12346, 12345

3.
```
type sir=array[1..100]of integer;

function interval(a:sir; n,k:integer):integer;
var c,i:integer;
begin
  c:=0;
  for i:=1 to n do
    if (a[i] >=a[1]) and (a[i] <=a[n])
      then c:=c+1;
  interval:=c;
end;
```

4.a. Se ovbserva ca numerele aflate inaintea unui numar k intr-un sir ordonat sunt toate mai mici decât numarul k. Bazându-ne pe această observație vom număra câte numere sunt mai mici decât primul numar din fisier.

Programul este eficient din punct de vedere al timpului de executie deoarece numerele sunt citite o singura data si imediat prelucrate şi nu se apeleaza la metode de ordonare.

Programul este eficient din punct de vedere al memoriei utilizate deoarece nu foloseşte structuri de date pentru a retine valorile ci imediat ce citeşte un număr îl şi prelucrează

4.b.
```
var f:text;
    n,i,c,nr,x:integer;

begin
  assign(f,'numere.txt'); reset(f);
  c:=1;
  read(f,n); read(f,nr);
  for i:=2 to n do begin
                     read(f,x);
                     if x<nr
                       then c:=c+1;
                   end;
  write(c);  close(f);
```

```
end.
```

Varianta 18:

1. b **2.** 11101, 11110, 11111

3.
```
function count(v:sir; n:byte):byte;
var i,ne,s:integer;
begin
  ne:=0; s:=0;
  for i:=1 to n do
    s:=s+v[i]
  for i:=1 to n do
    if v[i]>=s/n
      then ne:=ne+1;
  count:=ne;
end;
```

4.a. Numar câte numere sunt strict mai mari decât k şi, în acelaşi timp, verific dacă se gaseşte valoarea k printre numerele din fişier. In finalul programului afişez rezultatul (pozitia in care se gaseşte numarul sau mesajul adecvat după caz).

Programul este eficiane din punct de vedere al memoriei utilizate şi al timpului de execuție deoarece după fiecare citire a unui numar acesta este prelucrat ne mai fiind necesare alte citiri, nu se folosesc ordonari iar valorile nu sunt retinute in siruri (nu se folosesc structuri de date)

4.b.
```
var k,nr,c,x:integer;
    gasit:boolean;
    f:text;
begin
  assign(f,'numere.txt');  reset(f);
  write(' k= ');  read(k);
  c:=1;  gasit:=false;
  while not eof(f) do
    begin
      read(f,x);
      if x>k
        then c:=c+1;
      if x=k
        then gasit:=true;
    end;
  if gasit
    then write(c)
    else write('nu exista');
  close(f);
end.
```

Varianta 19:

1. a **2.** 10349, 10352, 10354

3.
```
type sir=array[1..100]of real;
procedure aranjare(var v:sir; n:byte);
var i,j:byte; aux:real;
begin
  i:=1; j:=n;
  while i<j do
    begin
      while (i<j) and (v[i]<0) do i:=i+1;
      while (i<j) and (v[j]>0) do j:=j-1;
      if i<j
        then begin
               aux:=v[i];
               v[i]:=v[j];
               v[j]:=aux;
             end;
    end;
end;
```

4.a. Citesc doua numere (câte unul din fiecare fişier) apoi pana nu am terminat de prelucrat numerele din ambele fişiere scriu la ecran cel mai mic dintre ultimele doua citite apoi mai citesc inca unul din fisierul corespunzător. Daca ambele numere sunt egale atunci afişez unul din ele si apoi citesc din fiecare fişier câte un număr.

Programul este eficient din punct de vedere al memoriei utilizate şi al timpului de execuţie deoarece citeşte o singură data numerele, nu foloseşte structuri de date şi nici ordonări / sortari.

4.b.
```
var f,g:text;
    n,m,i,j,x,y:longint;
begin
  assign(f,'nr1.txt'); reset(f);
  assign(g,'nr2.txt'); reset(g);
  read(f,n);  read(g,m);
  i:=1; j:=1;
  read(f,x);  read(g,y);
  while (i<=n) or (j<=m) do
    begin
      if (x=y)
        then begin
               write(x,' ');
               if not eof(f) then read(f,x)
                             else x:=maxlongint;
               if not eof(g) then read(g,y)
                             else x:=maxlongint;
               i:=i+1;   j:=j+1;
             end;
```

```
          if x<y
            then begin
                   write(x,' ');
                   if not eof(f) then read(f,x)
                                 else x:=maxlongint;
                   i:=i+1;
                 end;
          if x>y
            then begin
                   write(y,' ');
                   if not eof(g) then read(g,y)
                                 else y:=maxlongint;
                   j:=j+1;
                 end;
        end;
      close(f);
      close(g);
    end.
```

Varianta 20:

1. c **2.** 35789, 35679, 35678

3.
```
type sir=array[1..100]of real;
procedure nule(var v:sir; n:byte);
var i,j:byte; aux:real;
begin
  i:=1; j:=n;
  while i<j do
    begin
      while (i<j) and (v[i]<>0) do i:=i+1;
      while (i<j) and (v[j]=0) do j:=j-1;
      if i<j
        then begin
               aux:=v[i];
               v[i]:=v[j];
               v[j]:=aux;
             end;
    end;
end;
```

4.a. Citesc doua numere (câte unul din fiecare fişier) apoi pana nu am terminat de prelucrat numerele din ambele fişiere scriu la ecran unul dintre ultimele doua citite numai dacă sunt egale apoi mai citesc inca unul din fiecare fisier. Dacă numerele nu sunt egale atunci citesc din fişierul corespunzător un număr nou.

Programul este eficient din punct de vedere al memoriei utilizate şi al timpului de execuţie deoarece citeşte o singură data numerele, nu foloseşte structuri de date şi nici ordonări / sortari..

4.b.
```
type sir=array[1..100]of real;
var f,g:text;
    n,m,i,j,x,y:longint;
begin
  assign(f,'nr1.txt'); reset(f);
  assign(g,'nr2.txt'); reset(g);
  read(f,n);  read(g,m);
  i:=1; j:=1;
  read(f,x);  read(g,y);
  while (i<=n) or (j<=m) do
    begin
      if (x=y)
        then begin
               write(x,' ');
               if not eof(f) then read(f,x)
                             else x:=maxlongint;
               if not eof(g) then read(g,y)
                             else x:=maxlongint;
               i:=i+1;   j:=j+1;
             end;
      if x<y
        then begin
               if not eof(f) then read(f,x)
                             else x:=maxlongint;
               i:=i+1;
             end;
      if x>y
        then begin
               if not eof(g) then read(g,y)
                             else y:=maxlongint;
               j:=j+1;
             end;
    end;
  close(f);
  close(g);
end.
```

Varianta 21:

1. c **2.** 3

3.
```
function i_prim(n:integer):integer;
var p1,p2,i:integer;
    prim:boolean;
begin
  p1:=n+1;
  repeat
    p1:=p1-1;
    prim:=true;
    for i:=2 to p1 div 2 do
```

```
        if p1 mod i = 0
          then prim:=false;
    until prim;
    p2:=n-1;
    repeat
      p2:=p2+1;
      prim:=true;
      for i:=2 to p2 div 2 do
        if p2 mod i = 0
          then prim:=false;
    until prim;
    i_prim:=p2-p1;
  end;
```

4.a.
```
var a:array[1..10000]of integer;
    f:text;
    s,i,j,k,n,pn,un:integer;
    med:real;
begin
  assign(f,'bac.txt');  reset(f);
  s:=0;
  readln(f,n,k);
  for i:=1 to n do read(f,a[i]);
  for i:=1 to k do
    begin
      s:=s+a[i];
    end;
  med:=s / k;
  i:=1;
  for j:=2 to n-k+1 do
    begin
      s:=s-a[j-1]+a[j+k-1];
      if (s/k) > med
        then begin
               med:=s/k;
               i:=j;
             end;
    end;
  write(i);
  close(f);
end.
```

4.b. Citesc toate numerele într-un sir, fac suma primelor k numere apoi pentru rest uscad primul numar si adaug ultimul pentru aface o noua sumă. Dacă media aritmetica noua este mai mare decât cea initiala atunci salvez indicele noii medii aritmetice.

Metoda este eficientă din punct de vedere al timpului de executare deoarece numerele sunt prelucrate o singura data (intr-o singură parcurgere a şirului)

Varianta 22:

1. a **2.** ABACABA

3.
```
var k,n:integer;
function nz(n:integer):integer;
var nr2,nr5,x,i:integer;
begin
  nr2:=0;  nr5:=0;
  for i:=2 to n do
    begin
      x:=i;
      while x mod 2=0 do
        begin
          nr2:=nr2+1;
          x:= x div 2;
        end;
      while x mod 5=0 do
        begin
          nr5:=nr5+1;
          x:= x div 5;
        end;
    end;
  if nr2<nr5
    then nz:=nr2
    else nz:=nr5 ;
end;

begin
  write('k= '); read(k);
  n:=1;
  while nz(n)<k do n:=n+1;
  write(n,'!');
end.
```

4.
```
var a,b,i,n,p:integer;
    f:text;
begin
  assign(f,'bac.txt');  reset(f);
  read(f,n);
  for i:=1 to n do
    begin
      readln(f,a,b);
      p:=2;
      while p<b do p:=p*2;
      if p div 2 >=a
        then write(p div 2,' ')
        else write(0,' ');
    end;
  close(f);
end.
```

Varianta 23:

1. d **2.** 9

3.
```
type sir=array[1..100]of integer;
var n,i:integer;
    x:sir;

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

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

4.
```
var a,b:array[1..10]of integer;
    i,j,n:integer;
    ok:boolean;
    f:text;
begin
  assign(f,'bac.txt');  reset(f);
  read(f,n);
  for i:=1 to n do readln(f,a[i],b[i]);
  for i:=1 to n do
    begin
      ok:=true;
      for j:=1 to n do
        begin
          if (a[j]<a[i]) and (b[j]>b[i])
            then ok:=false;
          if (a[j]>a[i]) and (b[j]<b[i])
            then ok:=false;
          if (a[j]<a[i]) and (a[i]<b[j])
            then ok:=false;
          if (a[j]<b[i]) and (b[j]>b[i])
            then ok:=false;
        end;
      if ok
```

```
          then writeln(a[i],' ',b[i]);
      end;
    close(f);
  end.
```

Varianta 24:

1. a **2.** f(17)=3 f(22)=2

3.
```
type sir=array[1..100]of integer;
var x:sir;           M:real;
    n,i,min,max,sum:integer;

procedure P(n:integer; x:sir; var
mini,maxi,sum:integer);
begin
  mini:=x[1];
  for i:=1 to n do
    if x[i]<mini
      then mini:=x[i];
  maxi:=x[1];
  for i:=2 to n do
    if maxi<x[i]
      then maxi:=x[i];
  sum:=0;
  for i:=1 to n do
    sum:=sum+x[i];
end;

begin
  write(' n= '); read(n);
  write(' Introdu alorile: ');
  for i:=1 to n do
    read(x[i]);
  P(n,x,min,max,sum);
  M:=(sum-min-max)/(n-2);
  write(M:0:3);
end.
```

4.
```
var v:array[1..30000]of integer;
    n,a,b,i,min:integer;
    f:text;

begin
  assign(f,'bac.txt');   reset(f);
  read(f,n);
  for i:=1 to n do
    read(f,v[i]);
  read(f,a,b);
  min:=b+1;
  for i:=1 to n do
    if (v[i]<min) and (v[i]>=a)
      then min:=v[i];
```

```
    if min>b
      then write(' NU ')
      else write(min);
    close(f);
  end.
```

Varianta 25:

1. d **2.** a. f(16)=0 b. 95

3.
```
procedure f(n:integer; a:sir; var k:longint);
var i:integer; ok:boolean;
begin
  k:=0;  ok:=false;
  for i:=n downto 1 do
    if a[i] mod 2 = 0
      then begin
             k:=k*10+a[i];
             ok:=true;
           end;
  if not ok then k:=-1;
end;
```

4.a.
```
var f:text;
    x:real;
    p,n1,n2:longint;
begin
  assign(f,'numar.txt');   reset(f);
  read(f,x); p:=1;
  while x<>trunc(x) do begin
                         x:=x*10;
                         p:=p*10;
                       end;
  n1:=trunc(x);
  n2:=p;
  while n1<>n2 do
    if n1>n2
      then n1:=n1-n2
      else n2:=n2-n1;
  write(trunc(x) div n1,' ',p div n2);
  close(f);
end.
```

4.b. Programul inmulţeşte valoarea citita cu 10 până când aceasta devine întreagă, calculând simultan si puterea lui 10 corespunyatoare numărului de inmulţiri efectuate. La final determina c.m.m.d.c -ul celor doua numere intregi şi afişează numerele impărţite la cmmdc.

Programul este efficient deoarece nu foloseşte liste sau alte structuri pentru determiinarea valorilor solicitate iar acestea vor fi determinate in cel mai scurt timp

Varianta 26:

1. b **2.** 84211211

3.
```
var i,k,n:integer;
begin
  write(' n= '); read(n);
  write(' k= '); read(k);
  for i:=k downto 1 do
    write(n*i,' ');
end.
```

4.
```
type sir=array[1..100]of integer;
var v:sir;
    n,i,ant,na:integer;
    f:text;

procedure sterge(var v:sir;var n:integer; i,j:integer);
var k:integer;
begin
  for k:=i to n-j+i-1 do
    v[k]:=V[k+j-i+1];
  for k:=n+j-i+1 to n do
    v[k]:=0;
  n:=n-(j-i+1);
end;

begin
  assign(f,'numere.txt');   reset(f);
  read(f,n);
  for i:=1 to n do read(f,v[i]);
  ant:=v[1];  na:=0;
  i:=1;
  while i<=n do
    begin
      while (v[i]=ant) and (i<=n) do
         begin
           i:=i+1;
           na:=na+1;
         end;
      if na>1
        then begin
               ant:=v[i];
               sterge(v,n,i-na+1,i-1);
               i:=i-na+1;
               na:=1;
             end
        else ant:=v[i];
      i:=i+1;
    end;
  for i:=1 to n do write(v[i],' ');
end.
```

Varianta 27:

1. c **2.** 17263544444

3.
```
function nreal(x,y:integer):real;
var y1:real;
begin
  y1:=y;
  while y1>1 do
    y1:= y1 /10;
  nreal:=x+y1;
end;
```

4.a.
```
var f:text;
    un,pn:real;
    n,i,ne:integer;
begin
  assign(f,'numere.in');   reset(f);
  ne:=maxint;
  read(f,n);
  read(f,un);
  for i:=2 to n do
    begin
      pn:=un;
      read(f,un);
      if (ne>trunc(un)-trunc(pn)+1) and (pn=trunc(pn))
        then ne:=trunc(un)-trunc(pn)+1;
      if (ne>trunc(un)-trunc(pn)) and (pn<>trunc(pn))
        then ne:=trunc(un)-trunc(pn);
    end;
  write(ne);
  close(f);
end.
```

4.b. Știind că numerele sunt ordonate crescător vom determina numai pentru perechiile de elemente aflate pe poziții cunsecutive (restu nu mai este necesar) numărul de valori intregi. La final afişăm valoarea minimă determinata.

Programul este eficient din punct de vedere al memoriei utilizate fiindca nu se reţin toate elementele citite ci imediat ce un număr este citit este şi pelucrat.

Varianta 28:

1. a **2.** 5

3.
```
var a:array[1..10000]of real;
    n,i,c:integer;
```

```
    s:real;
begin
  write(' N= '); read(n);
  for i:=1 to n do
    begin
      write(' A[',i,']=');
      read(a[i]);
    end;
  s:=0;
  for i:=1 to n do
    s:=s+a[i];
  c:=0;
  for i:=1 to n do
    if a[i] = (s-a[i]) / (n-1)
      then c:=c+1;
  write(' c= ',c);
end.
```

4.

```
var a:array[1..10000]of integer;
    n,i,d1,d2,map:integer;
    gasit:boolean;
    f:text;

function primul(a:integer):integer;
var d:integer;
begin
  d:=2;
  while a mod d <> 0 do d:=d+1;
  primul:=d;
end;

begin
  assign(f,'numere.in'); reset(f);
  readln(f,n);
  for i:=1 to n do
    read(f,a[i]);
  gasit:=false;  map:=0;
  for i:=1 to n do
    begin
      d1:=primul(a[i]);
      d2:=a[i] div d1;
      if (primul(d1)=d1) and (primul(d2)=d2) and (a[i]>map)
        then begin
               gasit:=true;
               map:=a[i];
             end;
    end;
  if gasit
    then write(' DA ',map)
    else write(' NU ');
end.
```

Varianta 29:

1. c **2.** 1231210123

3.
```
function multiplii(a,b,c:integer):integer;
var i, m:integer;
begin
  m:=0;
  for i:=a to b do
    if i mod c = 0
      then m:=m+1;
  multiplii:=m
end;
```

4.a.
```
var a,b:array[1..100]of integer;
    n,m,i,j,s:integer;
    ok:boolean;    f:text;
begin
  assign(f,'numere.in'); reset(f);
  readln(f,n,m);
  for i:=1 to n do
    read(f,a[i]);
  for i:=1 to m do
    read(f,b[i]);
  i:=1; j:=1;
  ok:=true;
  while (j<=m) and (ok) do
    begin
      s:=0;
      while (s<b[j]) and (i<=n) do
        begin
          s:=s+a[i];
          i:=i+1;
        end;
      if s<>b[j]
        then ok:=false;
      j:=j+1;
    end;
  if ok
    then write(' DA ')
    else write(' NU ');
end.
```

4.b. Dupa citirea calori doi vectori incep analiaza acestora astfel: In variabila s adun valori ale primului sir (începând cu primul element) cât timp suma este mai mica decât valoarea curenta a sirului b (inițial prima). Cand aceasta conditie devine falsa verific dacă suma este diferită de valoarea curenta din şirul b. În caz afirmativ ma opresc fiindca nu se poate face reducerea. În caz negativ anulez suma, trec la urmatorul element in b si continu procedeul descris anterior.

Programul este eficient din punct de vedere al timpului de executare deoarece atat vectorul a cat si sirul b sunt parcursi o singura data ne mai fiind necesare intoarceri.

Varianta 30:

1. a **2.** 123443442344344

3.
```
type sir=array[1..100]of integer;

function suma(v:sir; n, i, j:integer):integer;
var s,i1:integer;
begin
  s:=0;
  for i1:=1 to i-1 do
    s:=s+v[i1];
  for i1:=j+1 to n do
    s:=s+v[i1];
  suma:=s;
end;
```

4.a.
```
var f:text;
    a:real;
    n,i,c,upi:integer;
begin
  assign(f,'numere.in'); reset(f);
  readln(f,n);
  c:=0; upi:=0;
  for i:=1 to n do
    begin
      read(f,a);
      if trunc(a)>upi
        then begin
               upi:=trunc(a);
               c:=c+1;
             end;
    end;
  write(' numar de intervale= ', c);
end.
```

4.b. Ma bazez pe faptul ca numerele sunt ordonate crescator si dupa ce citesc un numar determin imediat din ce interval face parte. Daca este vorba despre un interval nou il numar.

Programul este eficient din punct de vedere al timpului de executare deoarece numerele se citesc o singura data iar imediat după citire acestea sunt si prelucrate. Programul este eficient din punct de vedere al memoriei utilizate deoarece nu retine toate numerele ci numai cate unu, imediat după ctire facand si prelucrarea valorii.

Varianta 31:

1. c **2.** 681012108

3.
```
type sir:array[1..100]of integer;
function suma(x:sir; n,m:integer):integer;
var i:integer;
begin
  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 not ok;
  s:=0;
  for i:=1 to m do
    s:=s+x[i];
  suma:=s;
end;
```

4.
```
var f:text;
    x,y,x1,y1,i,n:integer;
begin
  assign(f,'numere.txt'); reset(f);
  readln(f,n);
  x1:=-100; y1:=100;
  for i:=1 to n do
    begin
      readln(f,x,y);
      if x1<x
        then x1:=x;
      if y1>y
        then y1:=y;
    end;
  if x1<=y1
    then write(x1,' ',y1)
    else write(0);
end.
```

Varianta 32:

1. c **2.** 164618

3.
```
function nr_prim(x:integer):integer;
var i,d:integer;
    prim:boolean;
begin
  prim:=false;
  while not prim do
    begin
      x:=x+1;
      prim:=true;
      for d:=2 to x div 2 do
        if x mod d=0
          then prim:=false;
    end;
  nr_prim:=x;
end;
```

4.
```
var f:text;
    min1, min2:integer;
    nr:longint;

begin
  assign(f,'numere.txt'); reset(f);
  min1:=1000; min2:=1000;
  while not eof(f) do
    begin
      read(f, nr);
      if (nr div 100 = 0) and (nr div 10<>0) and (nr<min2)
        then begin
               min1:=min2;
               min2:=nr;
             end;
    end;
  write(min1,' ',min2);
end.
```

Varianta 33:

1. b **2.** re(1)=10; re(14)=3

3.a.
```
function max_cif(x:sir; n:integer):integer;
var m,i:integer;
begin
    m:=-10000;
    for i:=1 to n do
      if( (x[i]>m) and (x[i]/1000=0) and (x[i]/100<>0) )
         then m:=x[i];
    max_cif := m;
end;
```

3.b.
```
type sir=array[1..100]of integer;
```

```
var n,m,i,j:integer;
    x,y:sir;
    f:text;
function max_cif(x:sir; n:integer):integer;
var m,i:integer;
begin
    m:=-10000;
    for i:=1 to n do
      if( (x[i]>m) and (x[i]/1000=0) and (x[i]/100<>0) )
         then m:=x[i];
    max_cif := m;
end;
begin
    m:=0;
    assign(f,'numere.txt '); reset(f);
    read(f,n);
    for i:=1 to n do
      begin
         for j:=1 to n do
           read(f,x[j]);
         m:=m+1;
         y[m]:=max_cif(x,n);
      end;
    m:=max_cif(y,n);
    if (m<>-10000)
       then write(m)
       else write(0);
end.
```

3.c. Programul citeste, pe rând, câte n elemente din fişier, apoi apeleaza functia max_cif memorând în continuare numai valoarea maximă. La final mai apelam o data functia pentru a determina cea mai mare valoare dintre toate numerele.

Programul este eficient fiindcă nu retine toate numerele ci pe rand numai cate o linie de n numere, astfel economisindu-se memorie.

Varianta 34:

1. a **2.** 3, 4 (oricare două nr consecutive, primul impar)

3.a.
```
function max(a:sir; n:integer):integer;
begin
    if(a[2]-a[1]>0)
      then max := a[n]
      else max := a[1];
end;
```

3.b. Deoarece numerele sunt în progresie aritmetica rezulta ca numerele sunt ordonate crescator sau descrescator în functie de ratie (pozitiva sau negativă). Daca ratia este pozitivă cel mai mare termen este ultimul din progresie, iar dacă

rația este negativă cel mai mare termen este primul. Verificând cum este rația aflam care este termenul ce trebuie întors de functie.

Metoda este eficientă deoarece nu verifică toate elementele progresiei ci se bazeaza pe proprietătile unei progresii aritmetice pentru a determina cea mai mare valoare.

3.c.
```
type sir=array[1..100]of integer;
var n,i,j,maxim,r:integer;
    x:sir;
    ok:boolean;
    f:text;
function max(a:sir;, n:integer):integer;
begin
    if(a[2]-a[1]>0)
      then max := a[n]
      else max := a[1];
end;
begin
    maxim:=0;
    assign(f, 'numere.txt '); reset(f);
    read(f,n);
    for i:=1 to n do
      begin
         for j:=1 to n do
           read(f,x[j]);
         r:=x[2]-x[1];
         ok:=true;
         for j:=1 to n-1 do
           if r <> x[j+1]-x[j]
             then ok:=false;
         if(ok)
           then if( maxim < max(x,n) )
                   then maxim:=max(x,n);
      end;
    write( maxim );
end.
```

Varianta 35:

1. c **2.** 4 2 -1 -3

3.
```
var n,i,x,nr:integer;
    f:text;
begin
    assign(f, 'numere.in'); reset(f);
    read(f,n);
    for i:=1 to n do
      begin
         read(f,nr); x:=nr;
```

```
           while x>9 do x := x div 10;
           if( x = nr mod 10 )
             then write(nr,' ');
        end;
  end.
```

4.a.
```
function sum(x:integer):integer;
var m:integer;
begin
  if (d<x/2)
    then begin
           d:=d+1;
           if ( x mod d = 0 )
             then begin
                    m:=d;
                    sum:= m + sum(x);
                  end
             else sum:= sum(x);
         end
    else sum:=0;
end;
```

4.b.
```
type sir=array[1..100]of integer;
var n,i,j,nr,aux,d:integer;
    a:sir;
function sum(x:integer):integer;
var m:integer;
begin
  if (d<x/2)
    then begin
           d:=d+1;
           if ( x mod d = 0 )
             then begin
                    m:=d;
                    sum:= m + sum(x);
                  end
             else sum:= sum(x);
         end
    else sum:=0;
end;
begin
    read(n);
    for i:=1 to n do
      begin
         read(nr);
         d:=1;
         a[i]:=sum(nr);
      end;
    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;
        for i:=1 to n do
          write(a[i], ' ');
    end.
```

Varianta 36:

1. b **2.** 2+3+7; 2+4+6

3.a.
```
function cifra(a:integer):integer;
begin
  if (a=0)
    then cifra:=0
    else begin
           while ( (a mod 2 = 1) and (a>0) ) do
              a := a div 10;
           if(a>0)
              then cifra:= a mod 10
              else cifra := -1;
         end;
end;
```

3.b.
```
type sir=array[0..9]of integer;
var n,i,j,nr,c:integer;
    a:sir;
    f:text;
function cifra(a:integer):integer;
begin
  if (a=0)
    then cifra:=0
    else begin
           while ( (a mod 2 = 1) and (a>0) ) do
              a := a div 10;
           if(a>0)
              then cifra:= a mod 10
              else cifra := -1;
         end;
end;
begin
    for i:=0 to 9 do a[i]:=0;
    assign(f, 'bac.in'); reset(f);
    read(f,n);
    for i:=1 to n do
      begin
         read(f,nr);
         c:=cifra(nr);
```

```
            if(c>=0) then a[c]:=a[c]+1;
          end;
        for i:=9 downto 0 do
          if  a[i] <> 0
             then for j:=1 to a[i] do
                    write(i);
    end.
```

3.c. Se citesc numerele pe rand şi imediat dupa ce a fost citit un număr se va determina cea mai mare cifra para. Cifra determinata va fi reținuta într-un şir de apariții apoi se va afişa fiecare cifra de câte ori apare.

Programul este eficient deoarece nu reține cele n cifre pare ci reține numatul de apariții ale lor astfel este necesar un sir de doar 10 elemente nu unul cu 15000 elemente.

Varianta 37:

1. c **2.** -11

3.
```
var n,i,j,k,x:integer;
a:array[1..100]of integer;
begin
    read(n,k);
    for i:=1 to n do
      read(a[i]);
    for i:=1 to k do
      begin
        x:=a[1];
        for j:=1 to n-1 do
          a[j]:=a[j+1];
        a[n]:=x;
      end;
    for i:=1 to n do
      write(a[i], ' ');
end.
```

4.a.
```
function nrdiv(x:integer):integer;
```

4.b.
```
var n,i,j,k,x,nr:integer;
    a:array[1..100]of integer;
    f:text
begin
    assign(f,'bac.in'); reset(f);
    read(f,n);
    for i:=1 to n do
      begin
         read(f,nr);
         if(nrdiv(nr) mod 2=0)
           then begin
```

```
                    if(x=0) then write(nr, ' ');
                    x:=nr;
                  end;
        end;
      write(x);
  end.
```

Varianta 38:

1. d **2.** 19

3.
```
function Del(x:longint, y:integer):longint;
var x1:longint;
begin
   x1:=0;
   while x>0 do
     begin
       if(x mod 10<=y)
         then x1:=x1*10+x mod 10;
       x := x div 10;
     end;
   x:=0;
   while x1>0 do
     begin
       x:=x*10+x1 mod 10;
       x1 := x1 div 10;
     end;
   if(x = 0)
     then Del:= -1
     else Del:= x;
end;
```

4.a.
```
procedure inter(var x,y:integer);
begin
   x := x + y;
   y := x - y;
   x := x - y;
end;
```

4.b.
```
procedure inter(var x,y:integer);
begin
   x := x + y;
   y := x - y;
   x := x - y;
end;
var n,i,j,k,x:integer;
    a:array[1..100]of integer;
    f:text;
begin
```

```
      assign(f, 'bac.in'); reset(f);
      read(f,n);
      for i:=0 to n do
        read(f,a[i]);
      for i:=0 to n-1 do
        for j:=i+1 to n do
           if(a[i]>a[j])
             then inter(a[i],a[j]);
      for i:=1 to n do
        write(a[i],'  ');
   end.
```

Varianta 39:

1. b **2.** 9

3.
```
var n,upc,nr,i:integer;
    ok:boolean;
    f:text;
begin
   assign(f,'bac.in'); reset(f);
   read(f,n);  upc:=0; ok:=true;
   for i:=1 to n do
     begin
       read(f,nr);
       if nr mod 2=0
         then if nr<=upc
                then ok:=false
                else upc:=nr;
     end;
   if (ok)
     then write(' DA ')
     else write(' NU ');
end.
```

4.a.
```
function pr(a:longint):boolean;
```

4.b.
```
var n:longint;
begin
   read(n);
   while n>99 do
     begin
       if( pr(n) )
         then write(n,' ');
       n := n div 10;
     end;
end.
```

Varianta 40:

1. c **2.** 12

3.
```
var n,up,ui,nr,i:integer;
    ok:boolean;
    f:text;
begin
   up:=0; ui:=9999; ok:=true;
   assign(f, 'bac.in'); reset(f);
   read(f,n);
   for i:=1 to n do
     begin
       read(f,nr);
       if(nr mod 2=0)
         then if(nr>=up)
                then up:=nr
                else ok:=false
         else if(nr<=ui)
                then ui:=nr
                else ok:=false;
     end;
   if (ok)
     then write(' DA ')
     else write(' NU ');
end.
```

4.a.
```
function pr(x:integer):boolean;
function sdiv(y:integer):integer;
```

4.b.
```
var i,n:integer;
begin
   read(n);
   for i:=1 to n do
     if( pr(sdiv(i)) )
       then write(i,' ');
end.
```

Varianta 41:

1. a **2.** f(7,2)=7; f(35,2)=5

3.
```
type sir=array[1..100]of integer;
function DIST(a:sir; n:integer):boolean;
var i,j:integer;
    ok:boolean;
begin
   ok:=true;
   for i:=1 to n-1 do
```

```
        begin
          for j:=i+1 to n do
            if(a[j]=a[i])
              then ok:=false;
          if( abs(a[i+1]-a[i]) <> 1 )
            then ok:=false;
        end;
      DIST := ok;
  end;
```

4.a.
```
var n:longint;
    i,nr,nam,max:integer;
    f:text;
begin
   max:=0; nam:=0;
   assign(f, 'numere.txt'); reset(f);
   read(f,n);
   for i:=1 to n do
     begin
       read(f,nr);
       if(max<nr)
         then begin
                max:=nr; nam:=1;
              end
         else if(max=nr)
                then nam:=nam+1;
     end;
   write(max,'  ',nam);
end.
```

4.b. Citesc pe rând câte un număr şi imediat ce l-am citit verific daca este mai mare decât maximul cunoscut pâna în acel moment. În caz afirmativ il reţin ca fiind noul maxim şi ca apare o data până acum. Dacă numărul citit este egal cu maximul atunci voi incrementa cu o unitate numărul de apariţii al maximului.

Programul este eficient deoarece numerele sunt citite o singură dată (din fişier) şi imediat prelucrate.

Varianta 42:

1. a **2.** `f(7)=6; f(100)=96;`

3.
```
type sir=array[1..100]of integer;
function P(a:sir; n,k:integer):integer;
var i,j,aux,sum:integer;
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;
      sum:=0;
      for i:=1 to k do
        sum := sum + a[i];
      P := sum;
   end;
```

4.a.
```
type sir=array[1..100]of integer;
var n,i:longint;
    nr:integer;
    f:text;  a:sir;
begin
   for i:=1 to 100 do a[i]:=0;
   assign(f,'numere.txt');   reset(f);
   read(f,n);
   for i:=1 to n do
     begin
       read(f,nr);
       a[nr]:=a[nr]+1;
     end;
   for i:=1 to n do
     if (a[i]=1)
       then write(i,'  ');
end.
```

4.b. Citim fiecare numar şi imediat ce am citit un numar vom numara apariția numărului. La final afişăm toate numarele care apar o singura dată.

Programul este eficien deoarece nu reține toate numerele iar pe2ntru afişatea acestora nu se foloseşte nicio metoda de ordonare,

---

Varianta 43:

1. a **2.** f(3)=6; f(10)=20;

3.
```
type sir=array[1..100]of integer;
function P(a:sir; n:integer):integer;
var i,j,sum:integer;
begin
   sum:=0;
   for i:=1 to n do
     if(a[i] mod 2=1) then sum := sum + a[i];
   P := sum;
end;
```

4.a.
```
type sir=array[0..9]of integer;
var n:longint;
```

```
        nr,i,j:integer;
        a:sir; f:text;
    begin
       for i:=0 to 9 do a[i]:=0;
       assign(f, 'numere.txt');  reset(f);
       read(f,n);
       for i:=1 to n do
         begin
           read(f,nr);
           a[nr]:=a[nr]+1;
         end;
       for i:=9 to 0 do
         for j:=1 to a[i] do
           write(i);
    end.
```

4.b. Programul citeşte cifrele din fişier. Imediat după citirea unei cifre din fisier va ncrementa numarul de apariţii al respectivei cifre. La final vom afişa cifrele în ordine descrescătoare, fiecare de câte ori apare.

Programul este eficient din punct de vedere al timpului de executare fiindcă nu execută operaţii de ordonare ale valorilor iar citirea se va executa o singura data (nu sunt necesare mai multe citiri ale datelor pentru rezolvarea problemei).

Varianta 44:

1. a **2.** 7

3.
```
type sir=array[1..100]of integer;
var n,i,npp:integer;
    a:sir;
begin
   read(n);
   for i:=1 to n do
     read(a[i]);
   npp:=0;
   for i:=1 to n do
     if( sqrt(a[i]) = trunc(sqrt(a[i])) )
       then npp:=npp+1;
   write(npp);
end.
```

4.a.
```
type sir=array[0..9]of integer;
var n:longint;
    nr,i:integer;
    f:text; a:sir;
begin
   assign(f, 'numere.txt'); reset(f);
   for i:=0 to 9 do a[i]:=0;
   read(f,n);
```

```
      for i:=1 to n do
        begin
          read(f,nr);
          while nr>0 do
            begin
              a[nr mod 10]:=a[nr mod 10]+1;
              nr := nr div 10;
            end;
          end;
      for i:=0 to 9 do
        if( a[i]>0 )
          then write(i,' ');
  end.
```

4.b. Programul citeşte, pe rând, fiecare numar, determină cifrele sale şi pentru fiecare cifra numara (incrementeaza) apariția acesteia. La final se afişează numai cifrele care conform şirului de apariții au appărut cel puțin o data în numerele citite.

Programul este eficient fiindcă pentru afişarea numerelor în ordine crescătoare nu se folosesc ordonări.

Varianta 45:

1. a **2.** `f(4)=3;  f(11)=5;`

3.
```
type sir=array[1..100]of real;
var i,j:integer; n:longint;
    a:sir;  f:text;
    ok:boolean
begin
   assign(f, 'numere.txt'); reset(f);
   read(f,n);
   for i:=1 to n do
     begin
       read(f,a[i]);
     end;
   for i:=1 to n do
     begin
       ok:=true;
       for j:=1 to n do
         if( (i<>j) and (a[i]=a[j]) )
           then ok:=false;
       if(ok)
         then write(a[i],' ');
     end;
end.
```

4.a.
```
type sir=array[0..100]of integer;
var n:longint;
    i,j,nr:integer;
```

```
        a:sir; f:text;
    begin
       assign(f, 'numere.txt');   reset(f);
       for i:=0 to 100 do a[i]:=0;
       read(f,n);
       for i:=1 to n do
         begin
           read(f,nr);
           a[nr]:=a[nr]+1;
         end;
       for i:=0 to 100 do
         for j:=1 to a[i] do
           write(i,' ');
    end.
```

4.b. Programul citeşte, pe rând, numerele infroduse în fişier. Imediat ce un număr a fost citit îl contorizează în sirul de apariţii. La final de afişează numerele care conform sirului de apariţii apr cel puţin o data. Numerele vor fi afişate de câte ori apar.

Programul este eficient fiindcă nu execută ordonari ale valorilor, ordonari care ar conuma mult timp.

Varianta 46:

1. c **2.** x=100

3.

```
type sir=array[0..100]of integer;
var nr,i,j,n,x,k,d:integer;
    f:text; a:sir;
begin
   x:=0;
   assign(f, 'bac.txt');  reset(f);
   read(k);
   read(f,n);
   for i:=1 to n do
     begin
       read(f,nr);
       d:=2;
       for j:=2 to nr div 2 do
         if(nr mod j=0)
           then d:=d+1;
       if(d>=k)
          then begin
                 x:=x+1;
                 a[x]:=nr;
               end;
     end;
   for i:=1 to x do
     write(a[i],' ');
end.
```

4.a.
```
procedure cifre(nr:longint; var nc,sc:integer);
```

4.b.
```
function sub(x:longint):boolean;
var nc,sc,c:integer;
    gasit:boolean;
begin
   gasit:=false;
   cifre(x,nc,sc);
   while x<>0 do
     begin
       c:=x mod 10;
       if( x = (sc-c)/(nc-1) )
         gasit:=true;
       x := x div 10;
     end;
   if( gasit )
     then sub := true
     else sub := false;
end;
```

Varianta 47:

1. d **2.** 6

3.
```
var nr,i,j,n,k,x:integer;
    f:text; ok:boolean;
begin
   ok:=false;
   assign(f, 'bac.txt'); reset(f);
   read(k);
   read(f,n);
   for i:=1 to n do
     begin
       read(f,nr);
       x:=nr;
       while(x mod 10<>k and x) do x := x div 10;
       if(x>0)
          then begin
                 write(nr,' ');
                 ok:=true;
               end;
     end;
   if( not ok)
     then write(' NU ');
end.
```

4.a.
```
procedure cif(nr:longint; var s:integer);
```

4.b.
```
type sir=array[0..25]of integer;
var n,i,smax,s:integer;
    a:sir;
begin
   read(n);
   for i:=1 to n do
     read(a[i]);
   smax=0;
   for i:=1 to n do
     begin
       cif(a[i],s);
       if( s>smax )
         then smax:=s;
     end;
   for i:=1 to n do
     begin
       cif(a[i],s);
       if( s=smax )
         then write(a[i],' ');
     end;
end.
```

Varianta 48:

1. c **2.** M1, M3, M2, M4

3.a.
```
function cmdiv(x,y:integer):integer;
begin
  while x <> y do
    if( x > y )
      then x := x - y
      else y := y - x;
  cmdiv := x;
end;
```

3.b.
```
type sir=array[0..100]of integer;
var n,i,m,a,b,aux:integer;
    s:sir;
function cmdiv(x,y:integer):integer;
begin
  while x <> y do
    if( x > y )
      then x := x - y
      else y := y - x;
  cmdiv := x;
end;
begin
   m:=0;
   read(a,b,n);
```

```
      if( a>b )
        then begin
               aux:=a;
               a:=b;
               b:=aux;
             end;
      for i:=a to b do
        if(cmdiv(n,i)=1)
          then begin
                 m:=m+1;
                 s[m]:=i;
               end;
      for i:=1 to m do
        write(s[i],' ');
  end.
```

4.
```
type sir=array[0..5000]of integer;
var n,i,j,na, aux:integer;
    a:sir;  f,g:text;
begin
   assign(f, 'bac.in'); reset(f);
   assign(g, 'bac.out'); rewrite(g);
   read(f,n);
   for i:=1 to n do
     read(f,a[i]);
   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;
   for i:=1 to n do
     begin
       na:=0;
       for j:=1 to n do
         if(a[i]=a[j])
           then na:=na+1;
       if(na=1)
         then write(g,a[i],' ');
     end;
   close(g);
end.
```

Varianta 49:

1. c **2.** 4

3.
```
type sir=array[0..30000]of integer;
```

```
var n,i,m,a,b,aux:integer;
    s:sir;
begin
   m:=0;
   read(a,b,n);
   if( a>b )
     then begin
            aux:=a;
            a:=b;
            b:=aux;
          end;
   for i:=a to b do
     if(i mod n=0)
       then begin
              m:=m+1;
              s[m]:=i;
            end;
   if(m=0)
     then write(' NU ')
     else for i:=1 to m do
            write(s[i],' ');
end.
```

4.a.
```
procedure cmax(a:integer; var b:integer);
```

4.b.
```
var nr,max,cif:integer;
    f:text;
begin
   assign(f, 'bac.txt'); reset(f);
   while not eof(f) do
     begin
       read(f,nr);
       cmax(nr,cif);
       if(cif>max)
         then max:=cif;
     end;
   write(max);
end.
```

Varianta 50:

1. b **2.** 332321

3.a.
```
function divxy(x,y:integer):boolean;
begin
  if( x mod y=0 )
    then divxy := true
    else divxy := false;
end;
```

3.b.
```
type sir=array[0..1000]of integer;
var n,i,m,a,b,aux:integer;
    s:sir;
function divxy(x,y:integer):boolean;
begin
  if( x mod y=0 )
    then divxy := true
    else divxy := false;
end;
begin
   read(a,b,n);
   if( a>b )
     then begin
            aux:=a;
            a:=b;
            b:=aux;
          end
   for i:=a to b do
     if(divxy(n,i))
       then begin
              m:=m+1;
              s[m]:=i;
            end;
   for i:=1 to m do
     write(s[i],' ');
end.
```

4.
```
type sir=array[0..5000]of integer;
var n,i,j,aux,na:integer;
    f,g:text;
    a:sir;
begin
   assign(f, 'bac.in'); reset(f);
   assign(g, 'bac.out'); rewrite(g);
   read(f,n);
   for i:=1 to n do
     read(f,a[i]);
   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;
   for i:=1 to n do
     begin
       na:=0;
       for j:=1 to n do
         if(a[i]=a[j])
```

```
            then na:=na+1;
         if (na>=2) and (a[i]<> a[i-1])
           then write(g,a[i],' ');
       end;
     close(g);
  end.
```

Varianta 51:

1. a **2.** 35280

3.a.
```
function dist2(xa,ya,xb,yb:integer):longint;
var l:longint;
begin
  l:=(xa-xb)*(xa-xb)+(ya-yb)*(ya-yb);
  dist2:=l;
end;
```

3.b.
```
function dist2(xa,ya,xb,yb:integer):longint;
var l:longint;
begin
  l:=(xa-xb)*(xa-xb)+(ya-yb)*(ya-yb);
  dist2:=l;
end;
var ax,ay,bx,by,cx,cy,dx,dy:integer;
    l1,l2,l3,l4:longint;
    ok:boolean;
begin
   read(ax,ay);
   read(bx,by);
   read(cx,cy);
   read(dx,dy);
   ok:=true;
   if( dist2(ax,ay,cx,cy) <> dist2(bx,by,dx,dy) )
     then ok:=false;
   l1 := dist2(ax,ay,bx,by);
   l2 := dist2(bx,by,cx,cy);
   l3 := dist2(cx,cy,dx,dy);
   l4 := dist2(dx,dy,ax,ay);
   if (l1<>l2) or (l2<>l3) or (l3<>l4) or (l4<>l1)
     then ok:=false;
   if(ok)
     then write(' DA ')
     else write(' NU ');
end.
```

4.
```
type sir=array[1..100]of integer;
var n,i,j:integer;
    f:text; ok:boolean;
```

```
      a:sir;
  begin
     assign(f,'date.in'); reset(f);
     read(f,n);
     for i:=1 to n do
       read(f,a[i]);
     for i:=1 to n do
       for j:=i+1 to n do
         if( ( (a[i] mod 2=0) and (a[j] mod 2=0) ) or
             ( (a[i] mod 2<>0) and (a[j] mod 2<>0) ) )
           then begin
                  writeln(a[i],' ',a[j]);
                  ok:=true;
                end;
     if( not ok )
       then write(0);
  end.
```

Varianta 52:

1. 120 **2.** `f(4)=10; f(100)=5050;`

3.a.
```
procedure dist(a:longint; var b:longint);
var c:array[0..9]of integer;   i:integer;
begin
   for i:=0 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;
   b:=0;
   for i:=0 to 9 do
     if(c[i]>=1)
       then b:=b+1;
end;
```

3.b.
```
type sir=array[1..100]of integer;
var n,i,max, nr:integer;
    f:text; x:longint;
    a:sir;
procedure dist(a:longint; var b:longint);
var c:array[0..9]of integer;   i:integer;
begin
   for i:=0 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;
      b:=0;
      for i:=0 to 9 do
        if(c[i]>=1)
          then b:=b+1;
  end;
  begin
     max:=0; nr:=0;
     assign(f, 'date.in');  reset(f);
     read(f,n);
     for i:=1 to n do
       read(f,a[i]);
     for i:=1 to n do
       begin
          dist(a[i],x);
          if( max<x )
            then max:=x;
       end;
     for i:=1 to n do
       begin
          dist(a[i],x);
          if( max=x )
            then write(a[i],' ');
       end;
  end.
```

4.

```
type sir=array[1..100]of integer;
var n,i,j,aux:integer;
    a:sir;  ok:boolean;
begin
   read(n);
   for i:=1 to n do
     read(a[i]);
   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;
   ok:=true;
   for i=1 to n do
     if(a[i]<>i)
       then ok:=false;
   if( ok )
     then write(' DA ')
     else write(' NU ');
end.
```

Varianta 53:

1. a **2.** 6

3.a.
```
function cmmdc(a,b:longint):longint;
```

3.b.
```
type sir=array[1..100]of integer;
var n,i,lmax,lc:integer;
    a:sir;
    f:text;
function cmmdc(a,b:longint):longint;
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);
   read(f,n);
   for i:=1 to n do
     read(f,a[i]);
   lmax:=0;   lc:=0;
   for i:=1 to n-1 do
     if( cmmdc(a[i],a[i+1])=1 )
        then lc:=lc+1
        else begin
               if(lmax<lc)
                  then lmax:=lc;
               lc:=0;
             end;
   write(lmax+1,'  ');
end.
```

4.
```
type sir=array[1..100]of integer;
var n,i,j,aux:integer;
    a:sir;
begin
   read(n);
   for i:=1 to n do
     read(a[i]);
   for i:=1 to n do
     begin
       aux:=a[1];
       for j:=2 to n do  a[j-1]:=a[j];
       a[n]:=aux;
       for j:=1 to n do
         write(a[j],' ');
       writeln;
     end;
```

```
end.
```

Varianta 54:

1. a **2.** 1 2 3 4 5 2 2 1 1 0

3.a.

```
##################################################
##################################################
###           VA FI REZOLVAT ULTERIOR            ###
##################################################
##################################################
```

3.b.

```
##################################################
##################################################
###           VA FI REZOLVAT ULTERIOR            ###
##################################################
##################################################
```

4.

```
##################################################
##################################################
###           VA FI REZOLVAT ULTERIOR            ###
##################################################
##################################################
```

Varianta 55:

1. a **2.** 20

3.a.

```
##################################################
##################################################
###           VA FI REZOLVAT ULTERIOR            ###
##################################################
##################################################
```

3.b.

```
##################################################
##################################################
###           VA FI REZOLVAT ULTERIOR            ###
##################################################
##################################################
```

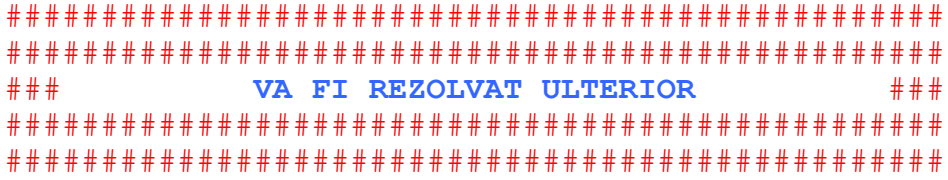
4.

Varianta 56:

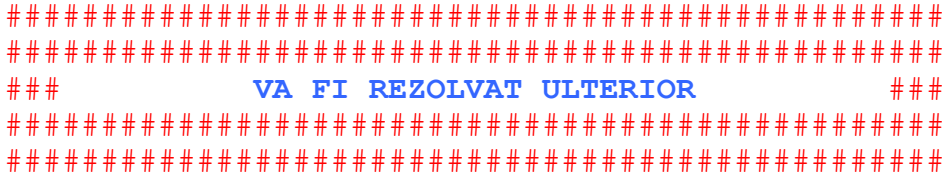
1. b **2.** 43

3.

4.a.



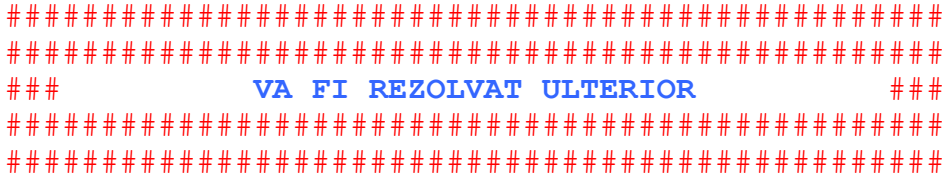
4.b.



Varianta 57:

1. a **2.** 3112

3.



4.a.

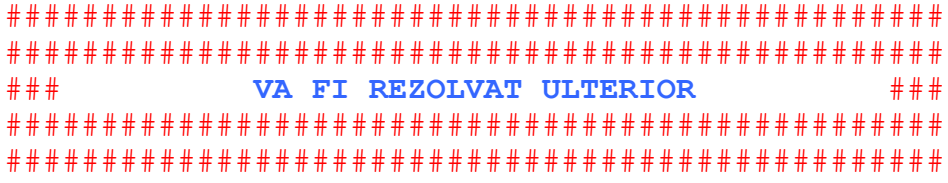
4.b.

Varianta 58:

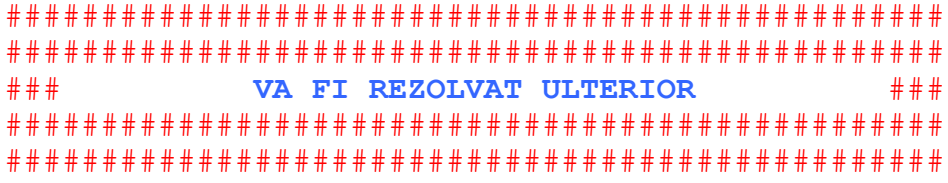
1. c **2.** 33

3.

4.a.



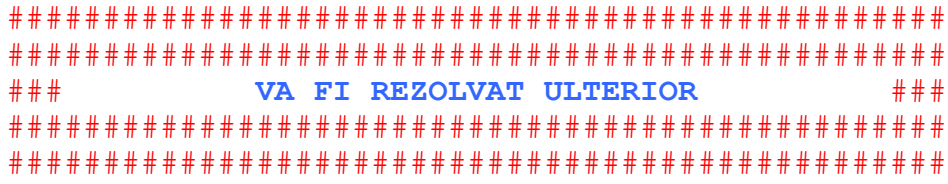
4.b.



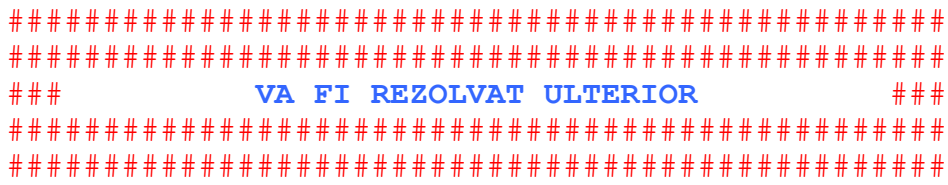
Varianta 59:

1. d **2.** 3452

3.



4.a.

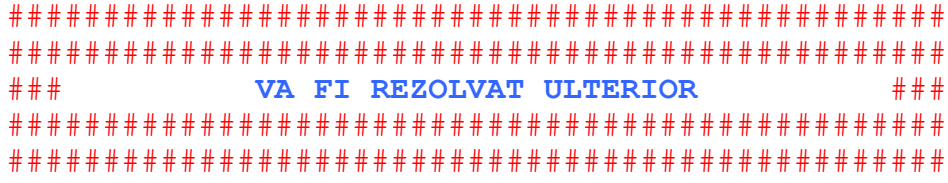


4.b.

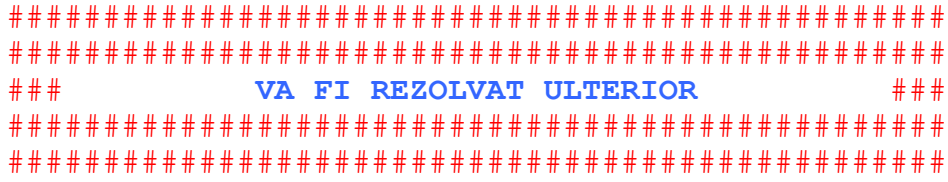
Varianta 60:

1. a **2.** 1604

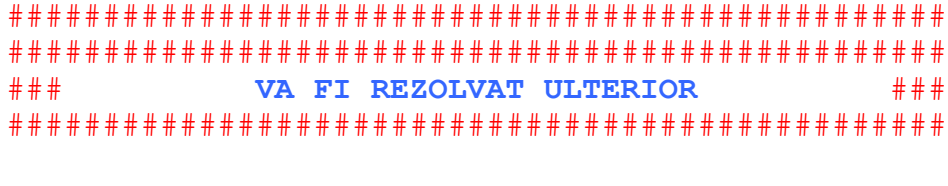
3.



4.a.



4.b.



```
##################################################
```

Varianta 61:

1. b **2.** 7

3.a.

```
##################################################
##################################################
###           VA FI REZOLVAT ULTERIOR            ###
##################################################
##################################################
```

3.b.

```
##################################################
##################################################
###           VA FI REZOLVAT ULTERIOR            ###
##################################################
##################################################
```

3.c.

```
##################################################
##################################################
###           VA FI REZOLVAT ULTERIOR            ###
##################################################
##################################################
```

Varianta 62:

1. a **2.** 9

3.

```
##################################################
##################################################
###           VA FI REZOLVAT ULTERIOR            ###
##################################################
##################################################
```

4.a.

```
##################################################
##################################################
###           VA FI REZOLVAT ULTERIOR            ###
```

```
##################################################
##################################################
```

4.b.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

Varianta 63:

1. 5 **2.** 15

3.a.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

3.b.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

3.c.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

Varianta 64:

1. d

2.
```
5*10=50
5*9=45
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.

```
##################################################
##################################################
###           VA FI REZOLVAT ULTERIOR            ###
##################################################
##################################################
```

4.a.

```
##################################################
##################################################
###           VA FI REZOLVAT ULTERIOR            ###
##################################################
##################################################
```

4.b.

```
##################################################
##################################################
###           VA FI REZOLVAT ULTERIOR            ###
##################################################
##################################################
```

Varianta 65:

1. a **2.** 222

3.a.

```
##################################################
##################################################
###           VA FI REZOLVAT ULTERIOR            ###
##################################################
##################################################
```

3.b.

```
##################################################
##################################################
###           VA FI REZOLVAT ULTERIOR            ###
##################################################
##################################################
```

3.c.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

Varianta 66:

1. c **2.** 48

3.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

4.a.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

4.b.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

Varianta 67:

1. a **2.** `alfa(4)=25;  alfa(6)=54;`

3.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

4.a.

```
##################################################
##################################################
###           VA FI REZOLVAT ULTERIOR            ###
##################################################
##################################################
```

4.b.

```
##################################################
##################################################
###           VA FI REZOLVAT ULTERIOR            ###
##################################################
##################################################
```

Varianta 68:

1. b **2.** 6 6 6 6 3

3.

```
##################################################
##################################################
###           VA FI REZOLVAT ULTERIOR            ###
##################################################
##################################################
```

4.a.

```
##################################################
##################################################
###           VA FI REZOLVAT ULTERIOR            ###
##################################################
##################################################
```

4.b.

```
##################################################
##################################################
###           VA FI REZOLVAT ULTERIOR            ###
##################################################
##################################################
```

Varianta 69:

1. a **2.** 15 5 9 3 1

3.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

4.a.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

4.b.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

Varianta 70:

1. d **2.** 11

3.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

4.a.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

4.b.

```
##################################################
##################################################
```

```
###           VA FI REZOLVAT ULTERIOR           ###
##############################################################
##############################################################
```

Varianta 71:

1. d **2.** 5

3.

```
##############################################################
##############################################################
###           VA FI REZOLVAT ULTERIOR           ###
##############################################################
##############################################################
```

4.a.

```
##############################################################
##############################################################
###           VA FI REZOLVAT ULTERIOR           ###
##############################################################
##############################################################
```

4.b.

```
##############################################################
##############################################################
###           VA FI REZOLVAT ULTERIOR           ###
##############################################################
##############################################################
```

Varianta 72:

1. b **2.** 111001

3.

```
##############################################################
##############################################################
###           VA FI REZOLVAT ULTERIOR           ###
##############################################################
##############################################################
```

4.a.

```
##############################################################
##############################################################
```

```
###           VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

4.b.

```
##################################################
##################################################
###           VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

Varianta 73:

1. a **2.** 137486

3.a.

```
##################################################
##################################################
###           VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

3.b.

```
##################################################
##################################################
###           VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

4.

```
##################################################
##################################################
###           VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

Varianta 74:

1. c **2.** `suma(8)=32;` `suma(11)=-60;`

3.

```
##################################################
```

```
##################################################
###           VA FI REZOLVAT ULTERIOR            ###
##################################################
##################################################
```

4.a.

```
##################################################
##################################################
###           VA FI REZOLVAT ULTERIOR            ###
##################################################
##################################################
```

4.b.

```
##################################################
##################################################
###           VA FI REZOLVAT ULTERIOR            ###
##################################################
##################################################
```

Varianta 75:

1. d **2.** -6 -2 0 5 10 7

3.a.

```
##################################################
##################################################
###           VA FI REZOLVAT ULTERIOR            ###
##################################################
##################################################
```

3.b.

```
##################################################
##################################################
###           VA FI REZOLVAT ULTERIOR            ###
##################################################
##################################################
```

4.

```
##################################################
##################################################
###           VA FI REZOLVAT ULTERIOR            ###
##################################################
##################################################
```

Varianta 76:

1. d **2.** 21

3.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

4.a.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

4.b.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

Varianta 77:

1. a **2.** 11

3.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

4.a.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

4.b.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

Varianta 78:

1. c **2.** 3

3.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

4.a.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

4.b.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

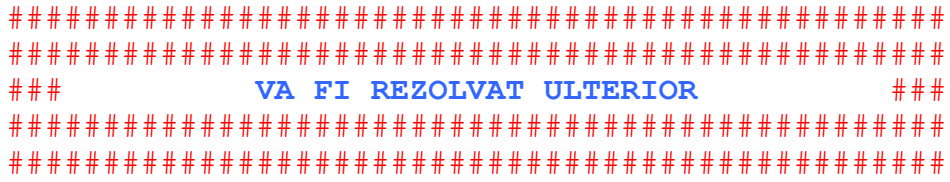
Varianta 79:

1. a **2.** 9

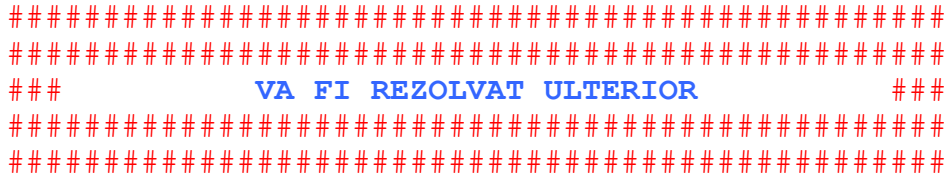
3.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

4.a.



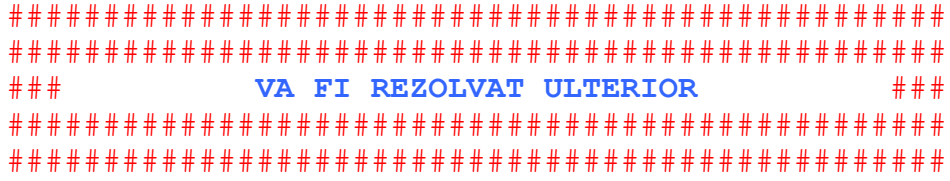
4.b.



Varianta 80:

1. b **2.** 11

3.



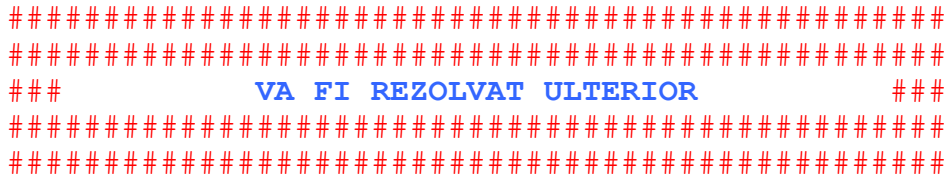
4.a.

4.b.

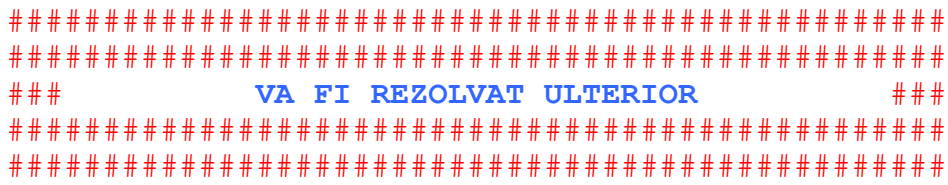
Varianta 81:

1. b **2.** 3

3.



4.a.

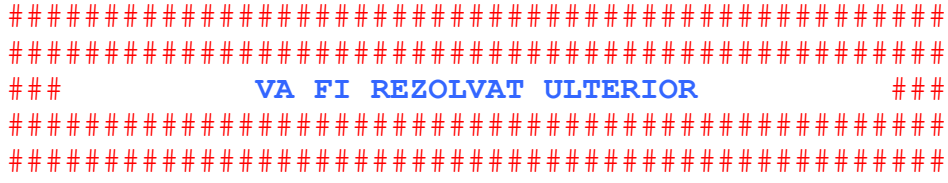


4.b.

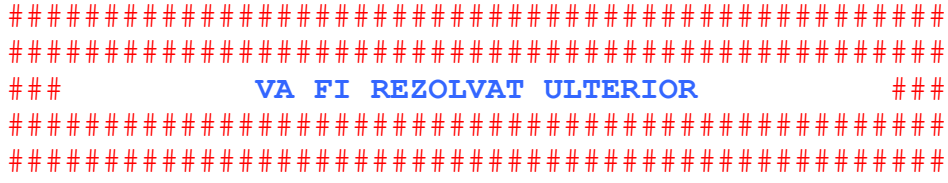
Varianta 82:

1. a **2.** 5

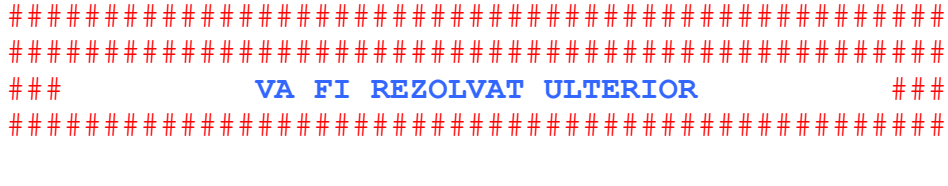
3.



4.a.



4.b.



```
################################################
```

Varianta 83:

1. b **2.** 11

3.

```
################################################
################################################
###          VA FI REZOLVAT ULTERIOR          ###
################################################
################################################
```

4.a.

```
################################################
################################################
###          VA FI REZOLVAT ULTERIOR          ###
################################################
################################################
```

4.b.

```
################################################
################################################
###          VA FI REZOLVAT ULTERIOR          ###
################################################
################################################
```

Varianta 84:

1. d **2.** 101

3.

```
################################################
################################################
###          VA FI REZOLVAT ULTERIOR          ###
################################################
################################################
```

4.

```
################################################
################################################
###          VA FI REZOLVAT ULTERIOR          ###
```

```
##################################################
##################################################
```

Varianta 85:

1. c **2.** 101

3.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

4.a.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

4.b.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

Varianta 86:

1. a **2.** ***##

3.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

4.a.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
```

```
##################################################
##################################################
```

4.b.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

Varianta 87:

1. b **2.** 43211234

3.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

4.a.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

4.b.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

Varianta 88:

1. a **2.** -2

3.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
```

```
##################################################
##################################################
```

4.a.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

4.b.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

Varianta 89:

1. c **2.** 30

3.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

4.a.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

4.b.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

Varianta 90:

1. a **2.** dcba

3.

```
##################################################
##################################################
###           VA FI REZOLVAT ULTERIOR            ###
##################################################
##################################################
```

4.a.

```
##################################################
##################################################
###           VA FI REZOLVAT ULTERIOR            ###
##################################################
##################################################
```

4.b.

```
##################################################
##################################################
###           VA FI REZOLVAT ULTERIOR            ###
##################################################
##################################################
```

Varianta 91:

1. d **2.** `i>0`

3.

```
##################################################
##################################################
###           VA FI REZOLVAT ULTERIOR            ###
##################################################
##################################################
```

4.a.

```
##################################################
##################################################
###           VA FI REZOLVAT ULTERIOR            ###
##################################################
##################################################
```

4.b.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

Varianta 92:

1. a **2.a.** 0 12 14 **2.b.** 27 şi 28

3.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

4.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

Varianta 93:

1. c **2.** 4443333332

3.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

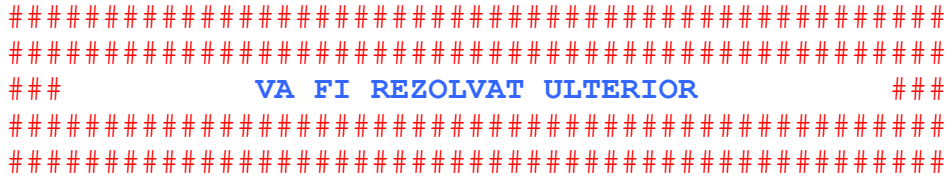
4.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

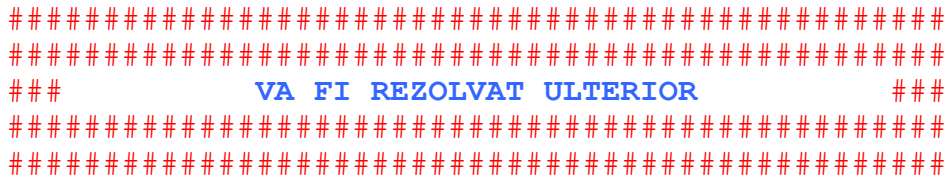
Varianta 94:

1. c **2.a.** 68 **2.b.** 8

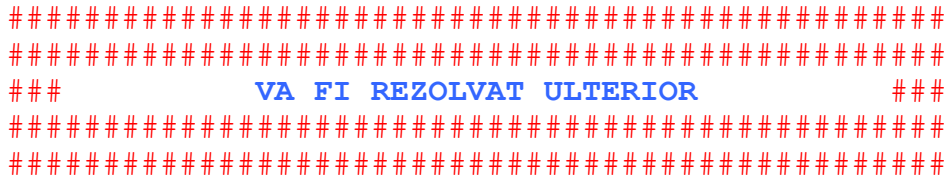
3.



4.a.



4.b.



Varianta 95:

1. b **2.** 11, 14, 17, 20, 22

3.

4.a.

4.b.

```
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

Varianta 96:

1. b **2.** 11, 12, 13, 17

3.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

4.a.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

4.b.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

Varianta 97:

1. b **2.** `agc, agf, agg`

3.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

4.a.

```
##################################################
```

```
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

4.b.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

Varianta 98:

1. c **2.** 5, 65, 25

3.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

4.a.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

4.b.

```
##################################################
##################################################
###          VA FI REZOLVAT ULTERIOR           ###
##################################################
##################################################
```

Varianta 99:

1. a **2.** wt, zx

3.

```
##################################################
```

```
##################################################
###           VA FI REZOLVAT ULTERIOR          ###
##################################################
##################################################
```

4.a.

```
##################################################
##################################################
###           VA FI REZOLVAT ULTERIOR          ###
##################################################
##################################################
```

4.b.

```
##################################################
##################################################
###           VA FI REZOLVAT ULTERIOR          ###
##################################################
##################################################
```

Varianta 100:

1. b **2.** 531024

3.

```
##################################################
##################################################
###           VA FI REZOLVAT ULTERIOR          ###
##################################################
##################################################
```

4.a.

```
##################################################
##################################################
###           VA FI REZOLVAT ULTERIOR          ###
##################################################
##################################################
```

4.b.

```
##################################################
##################################################
###           VA FI REZOLVAT ULTERIOR          ###
##################################################
##################################################
```