



!!! 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 greșeli și / sau pierderi survenite în urma folosirii lor!

Folosește rezolvările pe riscul tău !!!

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

Varianta 1:

1. b

2. 13245

3.

```
int main()
```

```
{
    int n, x, gasit=0;
    ifstream f('bac.txt');
    cout<<' n= '; cin>>n;
    while ( !f.eof() )
    {
        f>>x;
        if (x % n == 0)
        {
            cout<<x<<' ';
            gasit=1;
        }
    }
    f.close();
    if (gasit == 0 )
        cout<<" NU EXISTA ! ";
    return 0;
}
```

4.

```
int sub(int *v; int n; int a)
```

```
{
    int e=0, i;
    for(i=1; i<=n; i++)
        if (v[i]<a)
            e=e+1;
    return e;
}

int main()
{
    int a[101],i,n,gasit;
    cout<<" N= "; cin>>n;
    for(i=1; i<=n; i++)
    {
        cout<<" A["<<i<<"]= ";
        cin>>a[i];
    }
    do{
        ok=1;
        for (i=1; i<=n; i++)
            if (v[i]>v[i+1])
            {
                ok=0;
                v[i]=v[i]+v[i+1];
                v[i+1]=v[i]-v[i+1];
                v[i]=v[i]-v[i+1];
            }
    }
}
```

```
    }
while (!Ok);
ok=1;
for(i=2; i<=n; i++)
    if (sub(v,n,v[i])!=i-1)
        ok=0;
if (ok)
    cout<<"DA";
else cout<<" NU";
return 0;
}
```

Varianta 2:

1. d

2. 5310153

3. int main()

```
{
    int a[101], n, nr, i ,aux, sortat;
    ifstream f('nr.txt');
    n=0;
    while (!f.eof())
    {
        f>>nr;
        if (nr >0)
        {
            n=n+1;
            a[n]=nr;
        }
    }
    f.close();
    if (n==0) cout<<" NU EXISTA";
    else {
        do{
            sortat=1;
            for (i=1; i<=n-1; i++)
                if (a[i]>a[i+1])
                {
                    aux=a[i];
                    a[i]=a[i+1];
                    a[i+1]=aux;
                    sortat=0;
                }
        } while (!sortat);
        for (i=1; i<=n; i++)
            cout<<a[i]<<' ';
    };
    return 0;
}
```

```
4. int f(int a)
{
    int d=2, s=0, i;
    while (a>1)
    {
        while (a % d == 0)
        {
            a=a / d;
            s=s+1;
        }
        d=d+1;
    }
    return s;
}

int main()
{
    int n, n1;
    cout<<" n= "; cin>>n;
    if (n<10)
        { n1=2; n2=2; }
    else if (n<100)
        {
            n1=n % 10 *10 + n / 10;
            n2=2;
        }
    else {
        n1=n % 10 *10 + n / 10;
        n2=n1 % 10 *10 + n1 / 10;
    };
    if (f(n)==1) and (f(n1)==1) and (f(n2)==1)
        cout<<" DA ";
    else cout<<" NU ";
    return 0;
}
```

Varianta 3:

1. a

2. xyyy

```
3. int main()
{
    int a[101], n, nr, i, aux, sortat;
    ifstream f('nr.txt');
    n=0;
    while (!f.eof(f))
    {
        f>>nr;
        if ( (nr > 99) || (nr < -99) )
            {
```



```
        if (c==4)
            n1=n1*100+i*11;
        if (c==6)
            n1=n1*1000+i*111;
        if (c==8)
            n1=n1*10000+i*1111;
    };
}
n2=n1;
while (n1>0)
{
    n2=n2*10+n1 % 10;
    n1=n1 / 10;
};
cout<<n2<<endl;
return 0;
}
```

Varianta 4:

1. c

2. 01111

3. int main()

```
{
    int a[101], sortat, n, nr, i, aux;
    ifstream f('nr.txt');
    n=0;
    while (!f.eof(f))
    {
        f>>nr;
        if (nr < 100)
        {
            n=n+1;
            a[n]=nr;
        }
    }
    f.close();
    if (n==0)    cout<<" NU EXISTA";
    else {
        do{
            sortat=1;
            for (i=1; i<=n-1; i++)
                if (a[i]>a[i+1])
                {
                    aux=a[i];
                    a[i]=a[i+1];
                    a[i+1]=aux;
                    sortat=0;
                };
        } while (!sortat);
    }
}
```

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

4. int cif(long a; int b)
{
    int na=0;
    while (a>0)
    {
        if (a % 10 == b)
            na=na+1;
        a=a / 10;
    }
    return na;
}

int main()
{
    long n, c ,i, n1, n2;
    cout<<" n= "; cin>>n;
    n1=0;
    for (i=1; i<=9; i++)
    {
        c=cif(n,i);
        if ( (c>0) && (c % 2 != 0) )
        {
            n1=0; return 0;
        }
        else if (c>0) {
            if (c==2)
                n1=n1*10+i;
            if (c==4)
                n1=n1*100+i*11;
            if (c==6)
                n1=n1*1000+i*111;
            if (c==8)
                n1=n1*10000+i*1111;
        }
    }
    n2=n1
    while (n1>0)
    {
        n2=n2*10+n1 % 10;
        n1=n1 / 10;
    }
    cout<<n2<<endl;
    return 0;
}
```

Varianta 5:

1. d

2. 7*****

3.

```
int main()
```

```
{
    long n;
    ofstream f('nr.txt');
    cout<<" n= "; cin>>n;
    while (n>0)
    {
        f<<n<<' ';
        n=n / 10;
    }
    f.close();
    return 0;
}
```

4.

```
long f(long a)
```

```
{
    long d=2;
    while (a % d != 0) d++;
    return d;
}
int main()
{
    long a[101]; int i, n, gasit, ok, aux;
    cout<<" n= "; cin>>n;
    for (i=1; i<=n; i++)
    {
        cout<<" A["<<i<<"]= ";
        cin>>a[i];
    }
    do{
        ok=1;
        for(i=1; i<n; i++)
            if(a[i]>a[i+1])
            {
                ok=0;
                aux=a[i];
                a[i]=a[i+1];
                a[i+1]=aux;
            }
    }while( !ok );
    gasit=0;
    for (i=1; i<=n; i++)
        if (f(a[i]) == a[i])
        {
            gasit=1;
            cout<<a[i]<<' ';
        }
}
```



```
    }  
    if (!gasit)  
        cout<<" NU EXISTA! ";  
    return 0;  
}
```

Varianta 6:

1. b

2.7

```
3. int main()  
{  
    int i, n, s, a[101];  
    cout<<" n= "; cin>>n;  
    for (i=1; i<=n; i++)  
        {  
            cout<<"A["<<i<<"]="";  
            cin>>a[i];  
        }  
    s=0;  
    for( i=1; i<=n; i++) s=s+a[i];  
    for( i=n; i>=1; i--)  
        {  
            cout<<s<<endl;  
            s=s-a[i];  
        }  
    return 0;  
}
```

```
4. int main()  
{  
    long n1, n2, na=1;  
    ifstream fin("bac.txt");  
    fin>>n1;  
    while( fin>>n2 )  
        if (n1!=n2)  
            {  
                cout<<n1<<' '<<na<<' '  
                n1=n2; na=1;  
            }  
        else na=na+1;  
    cout<<n1<<' '<<na;  
    fin.close();  
    return 0;  
}
```

Varianta 7:

1. c

2. 126

```
-----  
  
3. int main()  
  {  
    int a[301], n, i, aux, ii, ip;  
    cout<<" n= "; cin>>n;  
    for(i=1; i<=3*n; i++)  
      {  
        cout<<"A["<<i<<"]="";  
        cin>>a[i];  
      }  
    ip=0; ii=0;  
    for( i=1; i<=n; i++)  
      if ( ( a[i] % 2 == 0 ) && ( ip == 0 ) )  
        ip=i;  
    for( i=3*n; i>=2*n+1; i--)  
      if ( ( a[i] % 2 == 1 ) && ( ii == 0 ) )  
        ii=i;  
    aux=a[ii];  
    a[ii]=a[ip];  
    a[ip]=aux;  
    for( i=1; i<=3*n; i++)  
      cout<<a[i]<<' '  
    return 0;  
  }  
  
4. long sub(long n)  
  {  
    long x=1, y;  
    while( x<n)  
      {  
        if (x<5)  
          x=x+1;  
        else x=2*x;  
      }  
    if (n>5)  
      return x / 2;  
    else return x;  
  }  
int main()  
  {  
    long s, n;  
    cout<<" s= "; cin>>s;  
    while (s>1)  
      {  
        n=sub(s);  
        cout<<n<<' '  
        s=s-n;  
      }  
    return 0;  
  }
```

Varianta 8:**1. b****2. 2**

1 2 3

3. `int sub(int n)`

```
{
    int s=0, i, x;
    for (i=1; i<=n; i++)
    {
        cin>>x;
        if (sqrt(x)==int(sqrt(x)))
            s=s+x;
    }
    return s;
}
```

4.a. `int main()`

```
{
    int a[101], b[101], i, j, n, m, ua;
    ofstream fout("bac.txt");
    cout<<" n= "; cin>>n;
    for (i=1; i<=n; i++)
    {
        cout<<"A["<<i<<"]="";
        cin>>a[i];
    }
    cout<<" m= "; cin>>m;
    for (j=1; j<=m; j++)
    {
        cout<<" B["<<j<<"]=" ";
        cin>>b[j];
    };
    i=1; j=1; ua=-a[1]-b[1];
    while ( (i<=n) && (j<=m) )
    {
        if (a[i]<b[j])
        {
            if (ua % 2 != a[i] % 2)
            {
                fout<<a[i]<<' ';
                ua=a[i];
            }
            i=i+1;
        }
        else {
            if (ua % 2 != b[j] % 2)
            {
                fout<<b[j]<<' ';
                ua=b[j];
            }
        }
    }
}
```

```
        j=j+1;
    }
}
while (i<=n)
{
    if (ua % 2 != a[i] % 2)
    {
        fout<<a[i]<<' ';
        ua=a[i];
    }
    i=i+1;
}
while (j<=m)
{
    if (ua % 2 != b[j] % 2)
    {
        fout<<b[j]<<' ';
        ua=b[j];
    }
    j=j+1;
}
fout.close();
return 0;
}
```

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 $a[i]$ și $b[j]$ numai daca difere de paritatea celei anterioare, avansând cu o poziție într-unul din șiruri.

Programul este eficient din punct de vedere al timpului de execuție deoarece citește 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. void sub(int n, int k)
{
    int i;
    for( i=n; i>=1; i--)
        cout<<n*k;
}
```

```
4.a. int main()
{
    int a[1000]; long x, i;
    memset(a,0,sizeof(int[1000]));
    ifstream fin("bac.txt");
```

```

while ( fin>>x )
    if (x<1000)
        a[x]=1;
i=999;
while ( (a[i]>0) && (i>0) ) i=i-1;
cout<<i<<' '; i=i-1;
while ( (a[i]>0) && (i>0) ) i=i-1;
cout<<i<<' ';
fin.close();
return 0;
}

```

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

Programul este eficient din punct de vedere al timpului de executare deoarece citește numai o dată datele din fișier.

Varianta 10:

1. a

2. 5

3. void sub(int n, int &a, int &b)

```

{
    int d, prim;
    a=0; b=0;
    while ( (a==0) || (b==0) )
    {
        prim=1;
        for (d=2; d<=n / 2; d++)
            if (n % d == 0)
                prim=0;
        if (prim)
            if (a==0)
                a=n
            else b=n;
        n=n-1;
    }
}

```

4.a) int main()

```

{
    int p[10000], t, v, c;
    memset(p,0,sizeof(int[10000]));
    ifstream fin("produse.txt");
    while (fin>>t>>v>>c)
        p[t]=p[t]+v*c;
    fin.close();
}

```

```

for (t=1; t<=9999; t++)
    if (p[t]>0)
        cout<<t<<' '<<p[t]<<endl;
return 0;
}

```

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

Programul este eficient deoarece nu folosește structuri de date pentru a reține toate elementele șirului (toate datele despre produse) și fișierul de intrare este citit o singură dată. În concluzie programul consumă puțină memorie și este rapid.

Varianta 11:

1. b

2. 1

3.a. Citim pe rând câte un număr din fișier. Dacă numărul nou citit este mai mare decât maximul de până atunci îl reținem ca fiind maxim. După fiecare citire afișăm maximul.

Programul este eficient din punct de vedere al timpului de execuție deoarece citește o singură dată numerele din fișier și nu reține toate numerele deci este eficient și din punct de vedere al spațiului de memorie utilizat.

```

3.b. int main()
{
    long n, i, x, xm=0;
    ifstream fin("numere.txt");
    fin>>n;
    for (i=1; i<=n; i++)
    {
        fin>>x;
        if (xm<x)
            xm=x;
        cout<<xm<<' ';
    }
    fin.close();
    return 0;
}

```

```

4. int sum(long x)
{
    long s=1, d;
    for (d=2; d<=x / 2; d++)
        if( x % d == 0 )
            s=s+d;
    return s+x;
}

```

```

int main()
{
    long n, i, x, c;
    cout<<" n= "; cin>>n;
    c=0;
    for (i=1; i<=n; i++)
        {
            cout<<" nr= "; cin>>x;
            if ( (sum(x)==x+1) && (x>1) )
                c=c+1;
        }
    cout<<" c= "<<c;
    return 0;
}

```

Varianta 12:**1. c**

2. 2+2+2+3; 2+2+5; 2+7

3.a) Mă bazez pe algoritmul interclasării a două șiruri. Până când nu am terminat de citi valorile din ambele fișiere citesc noi valori apoi afisez valoarea mai mică dintre cele două și divizibilă cu 5.

Programul este eficient deoarece nu reține toate elementele celor două fișiere ci numai câte o valoare, pe rând, deci consumă puțină memorie și percurge fișierele o singură dată deci este rapid.

3.b) `int main()`

```

{
    ifstream fin("nr1.txt");
    ifstream gin("nr2.txt");
    long x=0, y=0;
    while ( fin && gin )
        {
            if (x==y)
                {
                    fin>>x;
                    gin>>y;
                }
            else if (x<y)
                {
                    fin>>x ;
                    else gin>>y;
                }
            if ( (x<y) && (x % 5==0) )
                cout<<x<<' ';
            if ( (y<x) && (y % 5==0) )
                cout<<y<<' ';
        }
    while(fin)
        {
            if ( (x % 5==0) )

```

```
        cout<<x<<' ';
        fin>>x;
    }
    while(gin)
    {
        if (y % 5==0)
            cout<<y<<' ';
        gin>>y;
    }
    fin.close(); gin.close();
    return 0;
}
```

```
4. int main()
{
    long n, i, ne, x; int ok;
    cout<<" n= "; cin>>n;
    ne=0;
    for(i=1; i<=n; i++)
    {
        cin>>x;
        ok=1;
        while (x>9)
        {
            if (x % 10 != x / 10 % 10)
                ok=0;
            x=x / 10;
        }
        if (ok)
            ne=ne+1;
    }
    cout<<" Numere cu toate cifrele egale: "<<ne;
    return 0;
}
```

Varianta 13:**1. d****2. 6**

```
3. int main()
{
    int n, mg, p;
    cout<<" n= "; cin>>n;
    mg=0; p=0;
    while (p<n)
    {
        mg=mg+1;
        p=p+mg;
    }
}
```



```
        cout<<p-n+1;
        return 0;
    }

4. int main()
    {
        long nr, n;
        ifstream fin("bac.in");
        ofstream fout("bac.out");
        while (!fin.eof())
            {
                fin>>nr;
                n=0;
                while(nr)
                    {
                        if(nr%2==0)
                            n=n*10+nr%10;
                        nr /=10;
                    }
                while(n)
                    {
                        nr=nr*10+n%10;
                        n /=10;
                    }
                if (nr>0) fout<<nr<<' ';
            }
        fin.close(); fout.close();
        return 0;
    }
```

Varianta 14:

1. b

2. 72

```
3. int main()
    {
        int a[10], i, j, c, n;
        memset(a, 0, sizeof(int[10]));
        cout<<" n= "; cin>>n;
        cout<<" Introduceți cifrele: ";
        for(i=1; i<=n; i++)
            {
                cin>>c;
                a[c]=a[c]+1;
            };
        for (i=0; i<=9; i++)
            for (j=1; j<=a[i]; j++)
                cout<<i<<' ';
        return 0;
    }
```

```
}  
  
4. int main()  
{  
    int c, sp, x, s;  
    ifstream fin("bac.txt");  
    c=0; sp=0;  
    while (fin>>x)  
        {  
            cout<<x<<' '; c=c+1;  
            if (c % 5 == 0)  
                cout<<endl;  
            s=0;  
            while (x != 0)  
                {  
                    s=s+x % 10;  
                    x=x / 10;  
                };  
            if (s % 2 == 0)  
                sp=sp+1;  
        };  
    cout<<endl<<sp;  
    fin.close();  
    return 0;  
}
```

Varianta 15:

1. b

2. 85

```
3. int main()  
{  
    int i, j, nd, n, nr_div_max, nr_div;  
    cout<<" n= "; cin>>n;  
    nr_div_max=0; nr_div=0;  
    for (i=1; i<=n; i++)  
        {  
            nd=0;  
            for (j=1; i<=i; i++)  
                if (i % j == 0)  
                    nd=nd+1;  
            if (nd>nr_div)  
                {  
                    nr_div=nd;  
                    nr_div_max=i;  
                }  
        }  
    cout<<nr_div_max;  
    return 0;  
}
```

}

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. int main()
    {
        int pp, up, i, x, prim;
        ifstream fin("bac.in");
        up=0;
        while (fin>>x)
            {
                prim=1;
                for (i=2; i<=x / 2; i++)
                    if (x % i == 0)
                        prim=0;
                if (prim)
                    {
                        pp=up;
                        up=x;
                    }
            }
        fin.close();
        if (pp>0)
            cout<<pp<<' '<<up;
            else cout<<"Numere prime insuficiente";
        return 0;
    }

```

Varianta 16:

1. d

2. 77755, 77757, 77777

```

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

```

```

4. int main()
{
    int a[10], c, i, j; long n;
    memset(a, 0, sizeof(int[10]));
    ifstream fin("numere.txt");
    while (fin>>n)
    while (n>0)
        {
            c=n % 10;
            a[c]=a[c]+1;
            n=n / 10;
        }
    for (i=9; i>=0; i--)
        for (j=1; j<=a[i]; j++)
            cout<<i;
    fin.close();
    return 0;
}

```

Varianta 17:

1. c

2. 12347, 12346, 12345

```

3. int interval(int *a, int n, int k);
{
    int c=0, i;
    for (i=1; i<=n; i++)
        if ( (a[i]>=a[1]) && (a[i]<=a[n]) )
            c=c+1;
    return c;
}

```

4.a. Se observă că numerele aflate înaintea unui număr k într-un șir ordonat sunt toate mai mici decât numărul k . Bazându-ne pe această observație vom număra câte numere sunt mai mici decât primul număr din fișier.

Programul este eficient din punct de vedere al timpului de execuție deoarece numerele sunt citite o singură dată și imediat prelucrate și nu se apelează la metode de ordonare.

Programul este eficient din punct de vedere al memoriei utilizate deoarece nu folosește structuri de date pentru a reține valorile ci imediat ce citește un număr îl și prelucrează

```

4.b. int main()
{
    int n, i, c, nr, x;
    ifstream fin("numere.txt");
    c=1;
    fin>>n; fin>>nr;
}

```

```
for (i=2; i<=n; i++)
{
    fin>>x;
    if (x<nr)
        c=c+1;
};
cout<<c; fin.close();
return 0;
}
```

Varianta 18:**1. b****2. 11101, 11110, 11111**

```
3. int count(int *v, int n);
{
    int ne=0, s=0, i;
    for( i=1; i<=n; i++)
        s=s+v[i];
    for (i=1; i<=n; i++)
        if (v[i]>=s/n)
            ne=ne+1;
    return ne;
}
```

4.a. Numar câte numere sunt strict mai mari decât k și, în același timp, verific dacă se găsește valoarea k printre numerele din fișier. În finalul programului afișez rezultatul (poziția în care se găsește numărul sau mesajul adecvat după caz).

Programul este eficient din punct de vedere al memoriei utilizate și al timpului de execuție deoarece după fiecare citire a unui număr acesta este prelucrat și nu mai fiind necesare alte citiri, nu se folosesc ordonari iar valorile nu sunt reținute în siruri (nu se folosesc structuri de date)

```
4.b. int main()
{
    int k, nr, c, x, gasit;
    ifstream fin("numere.txt");
    cout<<" k= "; cin>>k;
    c=1; gasit=0;
    while ( fin>>x )
    {
        if (x>k)
            c=c+1;
        if (x==k)
            gasit=1;
    }
    if (gasit)
        cout<<c;
```

```

        else cout<<" nu exista ";
    fin.close();
    return 0;
}

```

Varianta 19:

1. a

2. 10349, 10352, 10354

```

3. void aranjare(double *v, int n)
{
    int i=1, j=n; double aux;
    while (i<j)
        {
            while ( (i<j) && (v[i]<0) ) i=i+1;
            while ( (i<j) && (v[j]>0) ) j=j-1;
            if (i<j)
                {
                    aux=v[i];
                    v[i]=v[j];
                    v[j]=aux;
                }
        }
}

```

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

Programul este eficient din punct de vedere al memoriei utilizate și al timpului de execuție deoarece citește o singură data numerele, nu folosește structuri de date și nici ordonări / sortari.

```

4.b. int main()
{
    long n, m, i, j, x, y;
    ifstream fin("nr1.txt");
    ifstream gin("nr2.txt");
    fin>>n; gin>>m;
    i=1; j=1;
    fin>>x; gin>>y;
    while ( (i<=n) || (j<=m) )
        {
            if ( x==y )
                {
                    cout<<x<<' ';
                    if (i<n) fin>>x;
                    else x=2147483647;
                }
        }
}

```

```

        if (j<m) gin>>y;
            else x=2147483647;
        i=i+1; j=j+1;
    }
    if ( x<y )
    {
        cout<<x<<' ';
        if (i<n) fin>>x;
            else x=2147483647;
        i=i+1;
    }
    if ( x>y )
    {
        cout<<y<<' ';
        if (j<m) gin>>y;
            else y=2147483647;
        j=j+1;
    }
}
fin.close(); gin.close();
return 0;
}

```

Varianta 20:

1. c

2. 35789, 35679, 35678

3. void nule(doble *v, int n)

```

{
    int i=1; j=n; double aux;
    while (i<j)
    {
        while ( (i<j) && (v[i]!=0) ) i=i+1;
        while ( (i<j) && (v[j]==0) ) j=j-1;
        if (i<j)
        {
            aux=v[i];
            v[i]=v[j];
            v[j]=aux;
        }
    }
}

```

4.a. Citesc doua numere (câte unul din fiecare fișier) apoi pana nu am terminat de prelucrat numerele din ambele fișiere 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 fișierul corespunzător un număr nou.

Programul este eficient din punct de vedere al memoriei utilizate și al timpului de execuție deoarece citește o singură dată numerele, nu folosește structuri de date și nici ordonări / sortari..

```
4.b. int main()
    {
        long n, m, x, y, i, j;
        ifstream fin("nr1.txt");
        ifstream gin("nr2.txt");
        fin>>n; gin>>m;
        i=1; j=1;
        fin>>x; gin>>y;
        while ( (i<=n) || (j<=m) )
            {
                if ( x==y )
                    {
                        cout<<x<<' ';
                        if (i<n) fin>>x;
                        else x=2147483647;
                        if (j<m) gin>>y;
                        else y=2147483647;
                        i=i+1; j=j+1;
                    }
                if ( x<y )
                    {
                        if (i<n) fin>>x;
                        else x=2147483647;
                        i=i+1;
                    }
                if ( x>y )
                    {
                        if (j<m) gin>>y;
                        else y=2147483647;
                        j=j+1;
                    }
            }
        fin.close(); gin.close();
        return 0;
    }
```

Varianta 21:**1. c****2. 3**

```
3. int i_prim(int n);
    {
        int p1, p2, i, prim;
        p1=n+1;
        do{
            p1=p1-1;
```



```

    prim=1;
    for (i=2; i<=p1 / 2; i++)
        if (p1 % i == 0)
            prim=0;
    }while (!prim);
    p2=n-1;
    do{
        p2=p2+1;
        prim=1;
        for (i=2; i<=p2 / 2; i++)
            if (p2 % i == 0)
                prim=0;
    }while (!prim);
    return p2-p1;
}

```

4.a. `int main()`

```

{
    int a[10000], s, i, j, k, n, pn, un;
    double med;
    ifstream fin("bac.txt");
    s=0;
    fin>>n>>k;
    for (i=1; i<=n; i++) fin>>a[i];
    for (i=1; i<=k; i++)
        {
            s=s+a[i];
        };
    med=s / k;
    i=1;
    for (j=2; j<=n-k+1; j++)
        {
            s=s-a[j-1]+a[j+k-1];
            if ( (s/k) > med )
                {
                    med=s/k;
                    i=j;
                }
        }
    cout<<i;
    fin.close();
    return 0;
}

```

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 șirului)

Varianta 22:

1. a

2. ABACABA

3. `int nz(int n)`

```
{
    int nr2=0, nr5=0, x, i;
    for (i=2; i<=n; i++)
    {
        x=i;
        while (x % 2==0)
        {
            nr2=nr2+1;
            x= x / 2;
        }
        while (x % 5==0)
        {
            nr5=nr5+1;
            x= x / 5;
        }
    }
    if (nr2<nr5)
        return nr2;
    else return nr5;
}
int main()
{
    int k, n;
    cout<<"k= "; cin>>k;
    n=1;
    while( nz(n)<k ) n=n+1;
    cout<<n<<'!'<<"are cel putin "<<k<<" zerouri";
    return 0;
}
```

4. `int main()`

```
{
    int a, b, i, n, p;
    ifstream fin("bac.txt");
    fin>>n;
    for (i=1; i<=n; i++)
    {
        fin>>a>>b;
        p=2;
        while (p<b) p=p*2;
        if (p / 2 >= a)
            cout<<p / 2<<' ';
        else cout<<0<<' ';
    };
}
```

```

    fin.close();
    return 0;
}

```

Varianta 23:**1. d****2. 9****3.** void shift(int *a, int n)

```

{
    int aux=a[1], i;
    for (i=1; i<=n-1; i++)
        a[i]=a[i+1];
    a[n]=aux;
}
int main()
{
    int x[100], n, i;
    cout<<" n= "; cin>>n;
    cout<<" Elementele lui x sunt.... ";
    for (i=1; i<=n; i++)
        cin>>x[i];
    for (i=n; i>=2; i--)
        shift(x,i);
    for (i=1; i<=n; i++)
        cout<<x[i]<<' ';
    return 0;
}

```

4. int main()

```

{
    int a[11], b[11], i, j, n, ok;
    ifstream fin("bac.txt");
    fin>>n;
    for(i=1; i<=n; i++) fin>>a[i]>>b[i];
    for(i=1; i<=n; i++)
    {
        ok=1;
        for (j=1; j<=n; j++)
        {
            if ( (a[j]<a[i]) && (b[j]>b[i]) )
                ok=0;
            if ( (a[j]>a[i]) && (b[j]<b[i]) )
                ok=0;
            if ( (a[j]<a[i]) && (a[i]<b[j]) )
                ok=0;
            if ( (a[j]<b[i]) && (b[j]>b[i]) )
                ok=0;
        }
    }
}

```

```

        if ( ok )
            cout<<a[i]<<' '<<b[i]<<endl;
    };
    fin.close();
    return 0;
}

```

Varianta 24:

1. a

2. $f(17)=3$ $f(22)=2$

3. void P(int n, int *x, int &mini, int &maxi, int &sum)

```

{
    int i;
    mini=x[1];
    for (i=1; i<=n; i++)
        if (x[i]<mini)
            mini=x[i];
    maxi=x[1];
    for (i=2; i<=n; i++)
        if (maxi<x[i])
            maxi=x[i];
    sum=0;
    for (i=1; i<=n; i++)
        sum=sum+x[i];
}
int main()
{
    int x[100], n, i, min, max, sum;
    double M;
    cout<<" n= "; cin>>n;
    cout<<" Introdu valorile: ";
    for (i=1; i<=n; i++)
        cin>>x[i];
    P(n,x,min,max,sum);
    M=double (sum-min-max)/double (n-2);
    cout<<setprecision(3)<<fixed<<M;
    return 0;
}

```

4. int main()

```

{
    int v[30000], n, a, b, i, min;
    ifstream fin("bac.txt");
    fin>>n;
    for (i=1; i<=n; i++)
        fin>>v[i];
    fin>>a>>b;
    min=b+1;
}

```

```
for(i=1; i<=n; i++)
    if ( (v[i]<min) && (v[i]>=a) )
        min=v[i];
if ( min>b )
    cout<<" NU ";
else cout<<min;
fin.close();
return 0;
}
```

Varianta 25:**1. d****2. a. $f(16)=0$ b. 95****3. void f(int n, int *a, long &k)**

```
{
    int i, ok;
    k=0; ok=0;
    for (i=n; i>=1; i--)
        if (a[i] % 2 == 0)
            {
                k=k*10+a[i];
                ok=1;
            };
    if (!ok) k=-1;
}
```

4.a. int main();

```
{
    double x; long p, n1, n2;
    ifstream f("numar.txt");
    fin>>x; p=1;
    while ( x!=int(x) )
        {
            x=x*10;
            p=p*10;
        }
    n1=int(x);
    n2=p;
    while (n1!=n2)
        if (n1>n2)
            n1=n1-n2;
        else n2=n2-n1;
    cout<<int(x) / n1<<" " <<p / n2;
    fin.close();
    return 0;
}
```

4.b. Programul înmulțește valoarea citită cu 10 până când aceasta devine întregă, calculând simultan și puterea lui 10 corespunzătoare numărului de înmulțiri efectuate. La final determină c.m.m.d.c -ul celor două numere întregi și afișează numerele împărțite la cmmdc.

Programul este eficient deoarece nu folosește liste sau alte structuri pentru determinarea valorilor solicitate iar acestea vor fi determinate în cel mai scurt timp.

Varianta 26:

1. b

2. 84211211

3.

```
int main()
```

```
{
    int i,k,n;
    cout<<" n= "; cin>>n;
    cout<<" k= "; cin>>k;
    for(i=k; i>=1; i--)
        cout<<n*i<<' ';
    return 0;
}
```

4.

```
void sterge(int *v, int &n, int i, int j)
```

```
{
    int k;
    for(k=i; k<=n-j+i-1; k++)
        v[k]=v[k+j-i+1];
    for(k=n+j-i+1; k<=n; k++)
        v[k]=0;
    n=n-(j-i+1);
}
```

```
int main()
```

```
{
    int v[100], i, n, ant, na;
    ifstream fin("numere.txt");
    fin>>n;
    for(i=1; i<=n; i++) fin>>v[i];
    ant=v[1]; na=0;
    i=1;
    while (i<=n)
    {
        while( (v[i]==ant) && (i<=n) )
        {
            i++;
            na++;
        }
        if (na>1)
        {
            ant=v[i];
        }
    }
}
```

Varianta 28:

1. a

2. 5

3.

```
int main()
```

```
{
    double a[10000],s; int n, i, c;
    cout<<" N= "; cin>>n;
    for(i=1; i<=n; i++)
        {
            cout<<" A["<<i<<"]=";
            cin>>a[i];
        }
    s=0;
    for(i=1; i<=n; i++)
        s=s+a[i];
    c=0;
    for(i=1; i<=n; i++)
        if ( a[i] == (s-a[i]) / (n-1) )
            c++;
    cout<<" c= "<<c;
    return 0;
}
```

4.

```
int primul(int a)
```

```
{
    int d=2;
    while ( a % d <> 0 ) d++;
    return d;
}
```

```
int main()
```

```
{
    int a[10000], n, i, d1, d2, map, gasit;
    ifstream fin("numere.in");
    fin>>n;
    for(i=1; i<=n; i++)
        fin>>a[i];
    gasit=0; map=0;
    for(i=1; i<=n; i++)
        {
            d1=primul(a[i]);
            d2=a[i] / d1;
            if( (primul(d1)=d1) && (primul(d2)=d2) && (a[i]>map) )
                {
                    gasit=1;
                    map=a[i];
                }
        }
}
```



```
if (gasit)
    cout<<" DA "<<map;
else cout<<" NU ";
return 0;
}
```

Varianta 29:

1. c

2. 1231210123

3. `int multiplii(int a, int b, int c)`

```
{
    int i, m=0;
    for(i=a; i<=b; i++)
        if( i % c == 0 )
            m++;
    return m;
}
```

4.a. `int main()`

```
{
    int a[100], b[100], i, j, s, n, m, ok;
    ifstream fin("numere.in");
    fin>>n>>m;
    for(i=1; i<=n; i++)
        fin>>a[i];
    for(i=1; i<=m; i++)
        fin>>b[i];
    i=1; j=1;
    ok=1;
    while( (j<=m) and (ok) )
    {
        s=0;
        while( (s<b[j]) and (i<=n) )
        {
            s=s+a[i];
            i++;
        }
        if( s<>b[j] )
            ok=0;
        j++;
    }
    if( ok )
        cout<<" DA ";
    else cout<<" NU ";
    return 0;
}
```

4.b. După citirea catoroi doi vectori încep analiza acestora astfel: În variabila s adun valori ale primului sir (începând cu primul element) cât timp suma este mai

mica decât valoarea curentă a șirului b (inițial prima). Când această condiție devine falsă verifică dacă suma este diferită de valoarea curentă din șirul b. În caz afirmativ mă opresc fiindcă nu se poate face reducerea. În caz negativ anulez suma, trec la următorul element în b și continui procedeul descris anterior.

Programul este eficient din punct de vedere al timpului de executare deoarece atât vectorul a cât și șirul b sunt parcurși o singură dată și nu mai fiind necesare întoarceri.

Varianta 30:

1. a

2. 123443442344344

```
3. int suma(int *v, int n, int i, int j)
{
    int il, s=0;
    for (il=1; il<=i; il++)
        s=s+v[il];
    for (il=j+1; il<=n; il++)
        s=s+v[il];
    return s;
}
```

```
4.a. int main()
{
    double a; int n, i, c, upi;
    ifstream fin("numere.in");
    fin>>n;
    c=0; upi=0;
    for(i=1; i<=n; i++)
    {
        fin>>a;
        if( int(a)>upi )
        {
            upi=int(a);
            c=c+1;
        }
    }
    cout<<" numar de intervale= "<< c;
    return 0;
}
```

4.b. Mă bazez pe faptul că numerele sunt ordonate crescător și după ce citesc un număr determin imediat din ce interval face parte. Dacă este vorba despre un interval nou îl număr.

Programul este eficient din punct de vedere al timpului de executare deoarece numerele se citesc o singură dată iar imediat după citire acestea sunt și prelucrate. Programul este eficient din punct de vedere al memoriei utilizate deoarece nu reține toate numerele ci numai câte unul, imediat după citire făcând și prelucrarea valorii.

Varianta 31:

1. c

2. 681012108

3.

```
int suma(int *x, int n, int m);
```

```
{
    int i, ok;
    do{
        ok=1;
        for(i=1; i<=n-1; i++)
            if( x[i]>x[i+1] )
                {
                    aux=x[i];
                    x[i]=x[i+1];
                    x[i+1]=aux;
                    ok=0;
                }
        while( ok );
        s=0;
        for(i=1; i<=m; i++)
            s=s+x[i];
        return s;
    }
```

4.

```
int main()
```

```
{
    int x, y, x1, y1, i, n;
    ifstream fin("numere.txt");
    fin>>n;
    x1=-100; y1=100;
    for(i=1; i<=n; i++)
        {
            fin>>x>>y;
            if( x1<x )
                x1=x;
            if( y1>y )
                y1=y;
        }
    if( x1<=y1 )
        cout<<x1<<' '<<y1;
    else cout<<0;
}
```

Varianta 32:

1. c

2. 164618


```
{
    if(x[1]-x[0]>0)
        return x[n-1];
    else return x[0];
}
int main()
{
    int n, x[100], i, j, maxim=0, r, ok;
    ifstream fin("numere.txt");
    fin>>n;
    for(i=0; i<n; i++)
    {
        for(j=0; j<n; j++)
            fin>>x[j];
        r=x[1]-x[0];
        ok=1;
        for(j=0; j<n-1; j++)
            if(r != x[j+1]-x[j])
                ok=0;
        if(ok)
            if( maxim < max(x,n) )
                maxim=max(x,n);
    }
    cout<< maxim;
    return 0;
}
```

Varianta 35:

1. c

2. 42-1-3

```
3. int main()
{
    int n, x, i, nr;
    ifstream fin("numere.in");
    fin>>n;
    for(i=0; i<n; i++)
    {
        fin>>nr; x=nr;
        while (x>9) x /= 10;
        if( x == nr%10 )
            cout<<nr<<" ";
    }
    return 0;
}
```

```
4.a. int sum(int x)
{
    static int d=1;
    if (d<x/2)
```

```
    {
        d++;
        if ( x%d == 0 )
            {
                int nr=d;
                return nr + sum(x);
            }
        else return sum(x);
    }
    else { d=1; return 0; }
```

```
4.b. int sum(int x)
{
    static int d=1;
    if (d<x/2)
        {
            d++;
            if ( x%d == 0 )
                {
                    int nr=d;
                    return nr + sum(x);
                }
            else return sum(x);
        }
    else { d=1; return 0; }
}

int main()
{
    int n, a[100], i, j, nr;
    cin>>n;
    for(i=0; i<n; i++)
        {
            cin>>nr;
            a[i]=sum(nr);
        }
    for(i=0; i<n-1; i++)
        for(j=i+1; j<n; j++)
            if( a[i] > a[j] )
                {
                    int aux=a[i];
                    a[i]=a[j];
                    a[j]=aux;
                }
    for(i=0; i<n; i++)
        cout<<a[i]<<" ";
    return 0;
}
```

```
3. int main()
{
    int n, a[100], i, j, k, x;
    cin>>n>>k;
    for(i=0; i<n; i++)
        cin>>a[i];
    for(i=0; i<k; i++)
    {
        x=a[0];
        for(j=0; j<n-1; j++)
            a[j]=a[j+1];
        a[n-1]=x;
    }
    for(i=0; i<n; i++)
        cout<<a[i]<<" ";
    return 0;
}
```

```
4.a. int nrdiv(int x);
```

```
4.b. int main()
{
    int n, a[100], i, j, k, x;
    ifstream fin("bac.in");
    fin>>n;
    for(i=0; i<n; i++)
    {
        fin>>nr;
        if(nrdiv(nr)%2==0)
        {
            if(x==0) cout<<nr<<" ";
            x=nr;
        }
    }
    cout<<x;
    return 0;
}
```

Varianta 38:

1. d 2. 19

```
3. long Del(long x, int y)
{
    long x1=0;
    while(x>0)
    {
        if(x%10<=y)
```

```
        x1=x1*10+x%10;
        x /= 10;
    }
    x=0;
    while(x1>0)
    {
        x=x*10+x1%10;
        x1 /= 10;
    }
    if(x == 0)
        return -1;
    else return x;
}
```

4.a. void inter(int &x, int &y)

```
{
    x = x + y;
    y = x - y;
    x = x - y;
}
```

4.b. void inter(int &x, int &y)

```
{
    x = x + y;
    y = x - y;
    x = x - y;
}
int main()
{
    int n, a[100], i, j, k, x;
    ifstream fin("bac.in");
    fin>>n;
    for(i=0; i<n; i++)
        fin>>a[i];
    for(i=0; i<n-1; i++)
        for(j=i+1; j<n; j++)
            if(a[i]>a[j])
                inter(a[i],a[j]);
    for(i=0; i<n; i++)
        cout<<a[i]<<" ";
    return 0;
}
```

Varianta 39:

1. b

2. 9

3. int main()
{

```
int n, upc, nr, ok, i;
ifstream fin("bac.in");
fin>>n; upc=0; ok=1;
for(i=0; i<n; i++)
{
    fin>>nr;
    if(nr%2==0)
        if(nr<=upc)
            ok=0;
        else upc=nr;
}
if (ok)
    cout<<" DA ";
else cout<<" NU ";
return 0;
}
```

4.a. int pr(long a);

4.b. int main()
{
 long n;
 cin>>n;
 while(n>99)
 {
 if(pr(n))
 cout<<n<<" ";
 n /= 10;
 }
 return 0;
}

Varianta 40:

1. c

2. 12

3. int main()
{
 int n, up=0, ui=9999, nr, ok=1, i;
 ifstream fin("bac.in");
 fin>>n;
 for(i=0; i<n; i++)
 {
 fin>>nr;
 if(nr%2==0)
 if(nr>=up)
 up=nr;
 else ok=0;
 else if(nr<=ui)


```

    }
    for(i=0; i<100; i++)
        if (a[i]==1)
            cout<<i<<" ";
    return 0;
}

```

4.b. Citim fiecare numar și imediat ce am citit un numar vom numara apariția numărului. La final afișăm toate numerele care apar o singura dată.

Programul este eficient deoarece nu reține toate numerele iar pentru afișarea acestora nu se folosește nicio metoda de ordonare,

Varianta 43:

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

```

3. int P(int *a, int n)
    {
        int i, j;
        int sum=0;
        for(i=0; i<n; i++)
            if(a[i]%2) sum += a[i];
        return sum;
    }

```

```

4.a. int main()
    {
        long n;
        int a[100],nr,i,j;
        memset(a, 0, sizeof(int[10]) );
        ifstream fin("numere.txt");
        fin>>n;
        for(i=0; i<n; i++)
            {
                fin>>nr;
                a[nr]++;
            }
        for(i=9; i>=0; i--)
            for(j=1; j<=a[i]; j++)
                cout<<i;
        return 0;
    }

```

4.b. Programul citește cifrele din fișier. Imediat după citirea unei cifre din fișier va incrementa numărul de apariții al respectivei cifre. La final vom afișa cifrele în ordine descrescătoare, fiecare de câte ori apare.

Programul este eficient din punct de vedere al timpului de executare fiindcă nu execută operații 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.

```
int main()
```

```
{
    int n, i, a[100], npp;
    cin>>n;
    for(i=0; i<n; i++)
        cin>>a[i];
    npp=0;
    for(i=0; i<n; i++)
        if( sqrt(a[i]) == int(sqrt(a[i])) )
            npp++;
    cout<<npp;
    return 0;
}
```

4.a.

```
int main()
```

```
{
    long n, a[10]; int nr, i;

    ifstream fin("numere.txt");
    memset(a, 0, sizeof(int[10]));
    fin>>n;
    for(i=0; i<n; i++)
    {
        fin>>nr;
        while(nr)
        {
            a[nr%10]++;
            nr /= 10;
        }
    }
    for(i=0; i<10; i++)
        if( a[i] )
            cout<<i<<" ";
    return 0;
}
```

4.b. Programul citește, pe rând, fiecare număr, determină cifrele sale și pentru fiecare cifra numara (incrementeaza) apariția acesteia. La final se afișează numai cifrele care conform șirului de apariții au apărut cel puțin o dată în numerele citite.

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

Varianta 46:**1. c****2. x=100**

```
3. int main()
{
    int nr, i, j, a[100], n, x=0, k;
    ifstream fin("bac.txt");
    cin>>k;
    fin>>n;
    for(i=0; i<n; i++)
    {
        fin>>nr;
        int d=2;
        for(j=2; j<=nr/2; j++)
            if(nr%j==0)
                d++;
        if(d>=k) a[x++]=nr;
    }
    for(i=0; i<x; i++)
        cout<<a[i]<<" ";
    return 0;
}
```

4.a. void cifre(long nr, int &nc, int &sc);

```
4.b. int sub(long x)
{
    int gasit=0, nc, sc;
    cifre(x,nc,sc);
    while(x!=0)
    {
        int c=x%10;
        if( x == (sc-c)/(nc-1) )
            gasit=1;
        x /= 10;
    }
    if( gasit )
        return 1;
    else return 0;
}
```

Varianta 47:**1. d****2. 6**

```
3. int main()
{
```

```
int nr, i, j, n, k, ok, x;

ifstream fin("bac.txt");
cin>>k;
fin>>n;
for(i=0; i<n; i++)
{
    fin>>nr;
    x=nr;
    while(x%10!=k && x) x /= 10;
    if(x) {cout<<nr<<" "; ok=1;}
}
if(!ok)
    cout<<" NU ";
return 0;
}
```

4.a. void cif(long nr, int &s);

4.b. int main()

```
{
    int n, i, a[25], smax, s;
    cin>>n;
    for(i=0; i<n; i++)
        cin>>a[i];
    smax=0;
    for(i=0; i<n; i++)
    {
        cif(a[i],s);
        if( s>smax )
            smax=s;
    }
    for(i=0; i<n; i++)
    {
        cif(a[i],s);
        if( s==smax )
            cout<<a[i]<<" ";
    }
    return 0;
}
```

Varianta 48:

1. c

2. M1, M3, M2, M4

3.a. int cmdiv(int x, int y)

```
{
    while( x != y )
        if( x > y )
```

```
        x -= y;
        else y -= x;
    return x;
}
```

```
3.b. int cmdiv(int x, int y)
{
    while( x != y )
        if( x > y )
            x -= y;
        else y -= x;
    return x;
}
int main()
{
    int n, i, s[100], m=0, a, b;
    cin>>a>>b>>n;
    if( a>b )
        {
            int aux=a;
            a=b;
            b=aux;
        }
    for(i=a; i<=b; i++)
        if(cmdiv(n,i)==1)
            s[m++]=i;
    for(i=0; i<m; i++)
        cout<<s[i]<<" ";
    return 0;
}
```

```
4. int main()
{
    int n, i, j, a[5000];
    ifstream fin("bac.in");
    ofstream fout("bac.out");
    fin>>n;
    for(i=0; i<n; i++)
        fin>>a[i];

    for(i=0; i<n-1; i++)
        for(j=i+1; j<n; j++)
            if(a[i]>a[j])
                {
                    int aux=a[i];
                    a[i]=a[j];
                    a[j]=aux;
                }
    for(i=0; i<n; i++)
        {
            int na=0;
```

```
        for(j=0; j<n; j++)
            if(a[i]==a[j])
                na++;
        if(na==1)
            fout<<a[i]<<" ";
    }
    fout.close();
    return 0;
}
```

Varianta 49:

1. c

2. 4

```
3. int main()
{
    int n, i, s[30000], m=0, a, b;
    cin>>a>>b>>n;
    if( a>b )
    {
        int aux=a;
        a=b;
        b=aux;
    }
    for(i=a; i<=b; i++)
        if(i%n==0)
            s[m++]=i;
    if(m==0)
        cout<<" NU ";
    else for(i=0; i<m; i++)
        cout<<s[i]<<" ";
    return 0;
}
```

4.a. void cmax(int a, int &b);

```
4.b. int main()
{
    int nr, max, cif;
    ifstream fin("bac.txt");
    while( fin>>nr)
    {
        cmax(nr, cif);
        if(cif>max)
            max=cif;
    }
    cout<<max;
    return 0;
}
```

Varianta 50:

1. b

2. 332321

```
3.a. int divxy(int x, int y)
    {
        if( x%y==0 )
            return 1;
        else return 0;
    }
```

```
3.b. int divxy(int x, int y)
    {
        if( x%y==0 )
            return 1;
        else return 0;
    }
int main()
{
    int n, i, s[1000], m=0, a, b;
    cin>>a>>b>>n;
    if( a>b )
        {
            int aux=a;
            a=b;
            b=aux;
        }
    for(i=a; i<=b; i++)
        if(divxy(n,i)==1)
            s[m++]=i;
    for(i=0; i<m; i++)
        cout<<s[i]<<" ";
    return 0;
}
```

```
4. int main()
    {
        int n, i, j, a[5000];
        ifstream fin("bac.in");
        ofstream fout("bac.out");
        fin>>n;
        for(i=0; i<n; i++)
            fin>>a[i];
        for(i=0; i<n-1; i++)
            for(j=i+1; j<n; j++)
                if(a[i]>a[j])
                    {
                        int aux=a[i];
```

```

        a[i]=a[j];
        a[j]=aux;
    }
    for(i=0; i<n; i++)
    {
        int na=0;
        for(j=0; j<n; j++)
            if(a[i]==a[j])
                na++;
        if(na>=2 && a[i]!= a[i-1])
            fout<<a[i]<<" ";
    }
    fout.close();
    return 0;
}

```

Varianta 51:**1. a****2. 35280**

3.a. `long dist2(int xa, int ya, int xb, int yb)`

```

{
    long l;
    l=(xa-xb)*(xa-xb)+(ya-yb)*(ya-yb);
    return l;
}

```

3.b. `long dist2(int xa, int ya, int xb, int yb)`

```

{
    long l;
    l=(xa-xb)*(xa-xb)+(ya-yb)*(ya-yb);
    return l;
}
int main()
{
    int ax,ay,bx,by,cx,cy,dx,dy;
    cin>>ax>>ay;
    cin>>bx>>by;
    cin>>cx>>cy;
    cin>>dx>>dy;
    int ok=1;
    if( dist2(ax,ay,cx,cy) != dist2(bx,by,dx,dy) )
        ok=0;
    long l1 = dist2(ax,ay,bx,by);
    long l2 = dist2(bx,by,cx,cy);
    long l3 = dist2(cx,cy,dx,dy);
    long l4 = dist2(dx,dy,ax,ay);
    if( l1!=l2 || l2!=l3 || l3!=l4 || l4!=l1 )
        ok=0;
    if(ok)

```

```
        cout<<" DA ";
        else cout<<" NU ";
        return 0;
    }

4. int main()
    {
        int n, i, j, a[100],ok;
        ifstream fin("date.in");
        fin>>n;
        for(i=1; i<=n; i++)
            fin>>a[i];
        for(i=1; i<n; i++)
            for(j=i+1; j<=n; j++)
                if( (a[i]%2==0 && a[j]%2==0) || (a[i]%2!=0 && a[j]%2!=0) )
                    {
                        cout<<a[i]<<" "<<a[j]<<endl;
                        ok=1;
                    }
        if( !ok )
            cout<<0;
        return 0;
    }
```

Varianta 52:

1. 120

2. $f(4)=10$; $f(100)=5050$;

3.a. void dist(long a, long &b)

```
{
    int c[10]={0,0,0,0,0,0,0,0,0,0};
    while(a)
        {
            c[a%10]++;
            a /= 10;
        }
    b=0;
    for(int i=0; i<=9; i++)
        if(c[i]>=1)
            b++;
}
```

3.b. void dist(long a, long &b)

```
{
    int c[10]={0,0,0,0,0,0,0,0,0,0};
    while(a)
        {
            c[a%10]++;
            a /= 10;
        }
```

```
    }
    b=0;
    for(int i=0; i<=9; i++)
        if(c[i]>=1)
            b++;
}
int main()
{
    int n, i, a[100], max=0, nr=0; long x;
    ifstream fin("date.in");
    fin>>n;
    for(i=0; i<n; i++)
        fin>>a[i];
    for(i=0; i<n; i++)
    {
        dist(a[i],x);
        if( max<x )
            max=x;
    }
    for(i=0; i<n; i++)
    {
        dist(a[i],x);
        if( max==x )
            cout<<a[i]<<" ";
    }
    return 0;
}
```

```
4. int main()
{
    int n, i, j, a[100];
    cin>>n;
    for(i=0; i<n; i++)
        cin>>a[i];
    for(i=0; i<n-1; i++)
        for(j=i+1; j<n; j++)
            if( a[i]>a[j] )
            {
                int aux= a[i];
                a[i]=a[j];
                a[j]=aux;
            }
    int ok=1;
    for(i=0; i<n-1; i++)
        if(a[i]!=i+1)
            ok=0;
    if( ok )
        cout<<" DA ";
    else cout<<" NU ";
    return 0;
}
```


Varianta 53:

1. a

2. 6

3.a. `long cmmdc(long a, long b);`

```
3.b. long cmmdc(long a, long b)
{
    while( a != b )
        if ( a>b ) a -= b;
        else b -= a;
    return a;
}
int main()
{
    int n, i, a[100], lmax, lc;
    ifstream fin("date.in");
    fin>>n;
    for(i=0; i<n; i++)
        fin>>a[i];
    lmax=0;   lc=0;
    for(i=0; i<n-1; i++)
        if( cmmdc(a[i],a[i+1])==1 )
            lc++;
        else{
            if(lmax<lc)
                lmax=lc;
            lc=0;
        }
    cout<<lmax+1<<" ";
    return 0;
}
```

4. `int main()`

```
{
    int n, i, j, a[100];
    cin>>n;
    for(i=0; i<n; i++)
        cin>>a[i];
    for(i=0; i<n; i++)
    {
        int aux=a[0];
        for(j=1; j<n; j++) a[j-1]=a[j];
        a[n-1]=aux;
        for(j=0; j<n; j++)
            cout<<a[j]<<" ";
        cout<<endl;
    }
}
```


1. d

2. 3452

3.

VA FI REZOLVAT ULTERIOR ###

#####

4.a.

VA FI REZOLVAT ULTERIOR ###

#####

4.b.

VA FI REZOLVAT ULTERIOR ###

#####

Varianta 60:

1. a

2. 1604

3.

VA FI REZOLVAT ULTERIOR ###

#####

4.a.

VA FI REZOLVAT ULTERIOR ###

#####

4.b.

#####

- 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          ###
```


#####

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 ###

#####

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 ###

#####

4.a.

VA FI REZOLVAT ULTERIOR ###

#####

4.b.

VA FI REZOLVAT ULTERIOR ###

#####

Varianta 80:

1. b

2. 11

3.

VA FI REZOLVAT ULTERIOR ###

#####

4.a.

VA FI REZOLVAT ULTERIOR ###

#####

4.b.

VA FI REZOLVAT ULTERIOR ###

#####

Varianta 81:

1. b

2. 3

3.

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

4.a.

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

4.b.

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

Varianta 82:

1. a

2. 5

3.

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

4.a.

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

4.b.

```
#####
```


VA FI REZOLVAT ULTERIOR ###

#####

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.

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 și 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          ###  
#####  
#####
```


