PROBLEME REZOLVATE ȘI EXPLICATE

INFORMATICĂ

MATEMATICĂ-INFORMATICĂ
CLASA A IX-A

MAZILESCU GEORGIAN
NEGRILESCU NICOLAE

PROBLEME REZOLVATE ȘI EXPLICATE

INFORMATICĂ

SPECIALIZAREA MATEMATCĂ-INFORMATICĂ,CLASA A IX-A

Autori: Mazilescu Georgian, Negrilescu Nicolae

ISBN 978-606-94763-7-6

Editura Evomind, 2020

https://evomind.org/

CUPRINS

CUVÂNT ÎNAINTE

Culegerea cuprinde o serie de probleme, organizate pe capitole, respectând structura programei școlare pentru clasele de matematică- informatică.

Un lucru important la această culegere este acela că la sfârșitul fiecărui capitol sunt prezentate rezolvările tuturor problemelor, programul c++ corespunzător și explicațiile necesare înțelegerii problemelor.

Lucrarea se adresează elevilor care se află în perioada de formare a bazelor gândirii algoritmice și a deprinderilor de programare în limbajul c++, a însușirii și folosirii raționamentului logico-deductiv, a găsirii de metode eficiente de rezolvare a problemelor.

Pentru elevi, cartea constituie un mijloc util de aprofundare a cunoștințelor teoretice și practice de programare a calculatoarelor, de formare a tehnicii și raționamentului algorithmic și de pregătire pentru concursurile de informatică.

ALGORITMI ELEMENTARI. ENUNȚURI

1. Se citește un număr natural n cu cel mult 4 cifre. Să se afiseze în ordine crescătoare toate numele naturale pare atât timp cât suma lor nu depășeste pe n.

   Ex: pentru n=15 se afiseaza 2 4 6.

2. Se citesc trei numere a,b,c. Să se afişeze valoarea maxima dintre cele trei numere.

   Exemplu: pentru a=10,b=7,c=20, se va afişa valoarea maxima 20.

3. Se consideră expresia E=$\begin{cases} x+1, & x>0 \\ x^2-2x+2, & x=0 \\ x^2-1, & x<0 \end{cases}$. Să se scrie un program care afişează valoarea expresiei pentru x citit de la tastatură.

   Exemplu: pentru x=7, se va afişa valoarea 8.

4. Se citesc n numere întregi Se cere să se afişeze media aritmetică a tuturor numerelor pare citite.

Exemplu: pentru n=6 şi numerele 8, 10, 17, 4, 9,107, se va afişa valoarea 7.33.

5. Se citesc n numere întregi. Să se scrie un program care calculează maximul dintre numerele negative.

Exemplu: pentru n=7 şi numerele 10, -4, 20, 12, -18, -7, 19, va fi afişată valoarea -4.

6. Să se scrie un program care rezolvă ecuaţia $ax^2+by^2=c$ în mulţimea numerelor întregi, pentru a,b,c citite de la tastatură.

Exemplu: pentru a=2, b=1, c=0, se afişeaza

7. Se citeşte un număr întreg n. Să se afişeze numărul obţinut prin eliminarea cifrelor impare.

Exemplu: pentru n=83276, va fi afişat numărul 826.

8. Se citeşte un număr întreg a. Să se scrie un program care aranjează descrescător cifrele lui a.

Exemplu: pentru a=6295, se va afişa numărul 9652.

9. Să se afişeze toate numerele prime dintr-un interval [a, b], unde aşi b sunt numere intregi citite de la tastatură.

Exemplu: pentru a=25, b=47, se vor afişa numerele: 29, 31, 37, 41.

10. Să se scrie un program care afişează toate numerele palindrom dintr-un interval [a, b], unde a,b se citesc de la tastatură.

Exemplu: pentru a=100, b=130, se afişează 101, 111, 121, 212, 222.

11. Se citeşte un număr natural n. Să se scrie un program care determină cel mai mare număr natural p astfel încât $2^p \leq n$.

Exemplu: pentru n=98 obţinem p=6 (2^6 =64 şi 2^7 = 128).

12. Să se scrie un program care afişează toate numerele de n (n<=8) cifre care se divid cu suma cifrelor lor.

Exemplu: pentru n=2, se afişează numerele: 10, 12,18, 20, 21, 24, 27,.....

13. Să se scrie un program care converteşte un număr intreg n din baza 10, în baza 2.

Exemplu: pentru n=56, se afişează 111000.

14. Se citeşte de la tastatură un număr natural n. Să se calculeze suma S=1 + 3 + 5+.....+2*n+1.

Exemplu: pentru n=5, se calculează suma 1+3+5+7+9+11=36.

15. Se citeşte un număr natural n. Să se scrie un program care generează primele n numere prime, care au inversele numere prime.

Exemplu: Pentru n=7 se vor afişa numerele 2, 3, 5, 7, 11, 13,17

16. Se citeşte un număr natural n. Să se afişeze următorul triunghi de numere:

1 2 3 4 ……..n

1 2 3 …….n-1

…….

1 2 3

1

17. Se citesc numitorul şi numărătorul unei fracţii a/b. Se cere să se afişeze fracţia simplificată.

Exemplu: pentru a=20 şi b=32, fracţia simplificată va fi: 5/8.

18. Să se scrie un Să se scrie un program care afişează primii n termeni ai următorului şir de numere: 1, 1, 2, 2, 1, 1, 2, 2, 3 ,3,3, 2, 2, 1,……

Exemplu: pentru n=3, se afişează 1 1, 2, 2, 1 1, 2, 2, 3, 3, 3, 2, 2, 1.

19. Se citesc n numere întregi. Să se afişeze numerele care au suma cifrelor divizibila cu 3.

    Exemplu: pentru n=5 şi numerele 3298, 127, 30, 567, 102 se afişează 30, 567, 102.

20. Să se scrie un program care să citească succesiv numere pana la introducerea valorii -1 şi care sa determine de câte ori apare cifra 0 în succesiunea de numere citite.

    Exemplu: dacă valoarea -1 se introduce după 6 numere citite şi numerele sunt 309, 129, 1004, 239, 12080 va fi afişată valoarea 5.

21. Se citeşte un număr natural n. Să se scrie un program care afişează toate numerele de la 1 la n care sunt pătrate perfecte.

    Exemplu: pentru n=50 vor fi afişate numerele 1, 4, 9, 16, 25, 36, 49.

22. Un număr este perfect dacă este egal cu suma divizorilor săi. Scrieţi un program care afişează toate numerele perfecte dintr-un interval [a, b], unde a şi b sunt citite de la tastatură.

Exemplu: pentru a=3, b=15, se afişează valoarea 6.

23. Se citesc n numere întregi. Să se determine cel mai mare număr prim dintre cele n numere citite.

    Exemplu: pentru n=6 şi numerele citite sunt 20, 19, 16, 29, 34, 78 cel mai mare număr prim este 29.

24. Se citeşte un număr întreg *n*. Să se scrie un program care roteşte numărul *n* la dreapta cu k poziţii.

    Exemplu: pentru n=45128 şi p=3, se afişează 12845.

25. Să se afişeze toate numerele palindrom de n cifre care au exact k cifre numere prime.

    Exemplu: n=3 şi k=3 se afişează 222, 232, 252, 272, 323, 333, 353, 373, 525, 535, 555, 575, 727, 737, 757, 777.

26. Sa se scrie un program care afişează toate numerele de 4 cifre având cifrele în ordine crescătoare şi suma lor egală cu s.

Exemplu: Un astfel de număr este 2456, pentru s=17.

Soluţie problema 1

Se citește numărul n. Se calculează suma numerelor pare cât timp este mai mică sau egală cu n. .

```
#include<iostream.h>
int main()
{
   int x,n,s;
  cin>>n;
  x=2;
  s=2;
  while(s<=n)
   { cout<<x<<" ";
     x=x+2;
     s=s+x;
   }
}
```

Soluţie problema 2

Se compară primele două numere a şi b şi se reţine valoarea maximă în variabila max, apoi se compară valoarea maximă obţinută cu cel de-al treilea număr c. Dacă max<c atunci maximul va fi c, altfel maximul rămâne cel obţinut din compararea primelor două numere.

```
#include <iostream.h>

int a,b,c,max;

int main()

{

   cout<<"a=";cin>>a;

   cout<<"b=";cin>>b;

   cout<<"c=";cin>>c;

   if(a>b)

         max=a;

   else

         max=b;

   if(max<c)

         max=c;

   cout<<"Valoarea maxima:"<<max;

   return 1;

}
```

Soluţie problema 3

Se citeşte x şi se atribuie variebilei E, expresia în funcţie de valoarea lui x, astfel: dacă x>0, E=x+1, dacă x=0 atunci E=x*x+2*x+2, altfel E=x*x+1.

```
#include <iostream.h>
int E,x;
int main()
{
   cout<<"x=";cin>>x;
   if(x>0)
         E=x+1;
   else
         if(x==0)
               E=x*x-2*x+2;
         else
               E=x*x-1;
         cout<<"Valoarea expresiei E="<<E;
         return 1;
}
```

Soluţie problema 4

Se vor citi pe rând cele n numere întregi în variabila a, şi în variabila s va fi reţinută suma lor. Pentru a calcula media aritmetică trebuie convertită suma celor n numere întregi la tipul de date float (float)s/n.

```
#include <iostream.h>
int a,n,i,s=0;
float ma;
int main()
{
   cout<<"Numarul de numere:";cin>>n;
   for(i=1;i<=n;i++)
   {
         cout<<"a=";cin>>a;
         s=s+a;
   }
   ma=(float)s/n;
   cout<<"Media aritmetica:"<<ma;
   return 1;
}
```

Soluţie problema 5

Se citesc cele n numere intregi in variabila a, valoarea max se iniţializează cu un număr negativ de tip int foarte mic, în cazul nostru am iniţializat a=-32000. Se testează dacă numărul introdus este negativ, şi în caz afirmativ se verifică dacă este mai mic decât valoarea reţinută în variabila max.

```
#include <iostream.h>
int a,n,max;
int main()
{  int i;
   max=-32000;
   cout<<"n=";cin>>n;
   for(i=1;i<=n;i++)
   {
         cout<<"a=";cin>>a;
         if(a<0 && a>max)
               max=a;
   }
   cout<<"Valoarea maxima negativa="<<max;
   return 1;
}
```

Soluţie problema 6

Se parcurg cu doi indici i şi j toate valorile de la 1 la c, şi pentru fiecare pereche (i, j) se verifică ecuaţia. În caz afirmativ perechea (i, j) este o soluţie a ecuaţiei şi se afişează pe ecran.

```
#include <iostream.h>
int a,b,c,i,j;
float x,y;
int main()
{
   cout<<"a=";cin>>a;
   cout<<"b=";cin>>b;
   cout<<"c=";cin>>c;
   for(i=0;i<=c;i++)
         for(j=0;j<=c;j++)
               if((a*i*i+b*j*j)==c)

   cout<<"("<<i<<","<<j<<") ";
   return 1;
}
```

Soluţie problema 7

Cât timp numarul citit n este diferit de zero, se extrage pe rând câte o cifră de la stânga la dreapta şi se verifică dacă este pară. În cazul în care această cifră este pară se adaugă la un nou număr n1, care iniţial este 0 (n1=(n%10)*p+n1).Noul număr format doar din cifre pare este n1.

```
#include <iostream.h>

int n,n1=0,p=1;

int main()

{

   cout<<"Introduceti numarul:";cin>>n;

   while(n!=0)

   {

         if(n%2==0)

         {

         n1=(n%10)*p+n1;

         p=p*10;

         }

         n=n/10;

   }

   cout<<n1;

   return 1;}
```

Soluţie problema 8

Se parcurg cifrele de la 9 la 0 şi se compară pe rând cu ultima cifră a numărului a. În caz afirmativ cifra se adaugă la un nou număr n care va conţine în final toate cifrele numărului a ordonate descrescător.

```
#include <iostream.h>

int a,i,a1,n=0;

int main()

{

   cout<<"a=";cin>>a;

   for(i=9;i>=0;i--)

   {

         a1=a;

         while(a1!=0)

         {

               if(a1%10==i)

            n=n*10+a1%10;

               a1=a1/10;

         }

   }

   a=n;

   cout<<a;
```

```
    return 1;
}
```

Soluţie problema 9

Dacă b>a atunci intervalul introdus este incorect, altfel se parcurg toate numerele din interval şi pentru fiecare număr se verifică proprietatea de număr prim. În caz afirmativ se efişează pe ecran.

```
#include <iostream.h>

int a,b,i,prim,x;

int main()

{

   cout<<"Introduceti                capetele
intervalului:";

   cout<<"a=";cin>>a;

   cout<<"b=";cin>>b;

   if(a>b)

         cout<<"Interval incorect";

   else

   {

         for(i=a;i<=b;i++)

         {

               x=i;

               prim=1;
```

```
                for(int j=2;j<=x/2;j++)
                        if(x%2==0)
                                prim=0;
                        if(prim)
                                cout<<i<<" ";
            }
    }
    return 1;
}
```

Soluţie problema 10

Se citesc capetele intervalului în variabilele a şi b şi se verifică dacă a<=b. În caz afirmativ se parcurg toate numerele din interval, se determină inversul fiecărui număr şi se compară cu numărul iniţial. Dacă cele două numere sunt egale, atunci numărul se afişează pe ecran.

```
#include <iostream.h>

int a,b,nr,i,x;

int main()

{

   cout<<"Introduceti                capetele
intervalului:\n";

   cout<<"a=";cin>>a;

   cout<<"b=";cin>>b;

   if(b<a)

         cout<<"Interval incorect";

   else

   {

         for(i=a;i<=b;i++)

         {

               int inv=0;
```

```
                x=i;
                while(x)
                {
                        inv=inv*10+x%10;
                        x=x/10;
                }
                if(inv==i)
                        cout<<i<<" ";
        }
    }
    return 1;
}
```

Soluţie problema 11

Vom folosi funcţia pow din biblioteca math.h pentru a ridica un număr la o putere dată. Pornim cu un indice i de la 1 şi testăm pe rând inegalitatea $2^i < n$ până când aceasta devine falsă. Afişăm valoarea i+1.

```
#include <iostream.h>
#include <conio.h>
#include <math.h>
int n,p,i;
int main()
{
   cout<<"n=";cin>>n;
         i=1;
   while(pow(2,i)<n) i++;
   p=i-1;
   cout<<p;
   getch();
   return 1;
}
```

Soluţie problema 12

Declarăm cele două capete ale intervalului a şi b de tip long int, deoarece pot fi numere de maxim 8 cifre. Utilizăm funcţia matematică pow, pentru a ridica un număr la o putere, pentru determinarea capetelor intervalului. Intervalul în care se caută numere de n cifre este [10^{n-1}, 10^{n} -1]. Se parcurg toate valorile din interval, se calculează suma cifrelor pentru fiecare număr iar acelea care se divid cu suma cifrelor lor se afişează pe ecran.

```
#include <iostream.h>
#include <math.h>
long int n,i,a,b,x;
int s;
int main()
{
   cout<<"n=";cin>>n;
   a=pow(10,n-1);
   b=pow(10,n)-1;
   for(i=a;i<=b;i++)
   {
         s=0;
         x=i;
```

```
            while(x)
            {
                  s+=x%10;
                  x=x/10;
            }
            if(i%s==0)
                  cout<<i<<" ";
      }
      return 1;
}
```

Soluţie problema 13

Vom forma un nou număr din resturile obţinute prin împărţiri succesive ale numărului n la 2. Folosim o variabilă p, care iniţial va fi 1, care va reprezenta ordinul de multiplicitate al restului obţinut prin împărţirea numărului n la 2. Numărul transformat în baza 2 se obţine astfel nr=(n%2)*p+nr, cât timp n este diferit de 0.

```
#include <iostream.h>
int n,p,nr=0;
int main()
{
   cout<<"n=";cin>>n;
   p=1;
   while(n!=0)
   {
         nr=(n%2)*p+nr;
         p=p*10;
         n=n/2;
   }
   cout<<"Numarul in baza 2="<<nr;
   return 1;
}
```

Soluţie problema 14

Fiecare termen al sumei este dat de 2*i+1, i=1,n. se parcurg din 2 în 2 numerele de la 1 la 2*n+1, la fiecare pas se calculează s=s+i;

```
#include <iostream.h>
int n,i,s=0;
int main()
{
   cout<<"n=";cin>>n;
   for(i=1;i<=2*n+1;i+=2)
         s=s+i;
   cout<<"Suma="<<s;
   return 1;
}
```

Soluţie problema 15

Vom folosi o variabilă nr în care vom reţine câte numere prime care au şi inversele prime am găsit până la un moment dat. Căutam un numar prim, după care verificăm dacă şi numărul care se formează cu cifrele citite de la coadă la cap este prim. Dacă este prim atunci creştem valoarea variabilei nr cu 1 şi afişăm numărul găsit.

```
#include <iostream.h>
#include <conio.h>
int n,prim,i,nr=0,j;
int main()
{

   cout<<"n=";cin>>n;
   i=2;
   while(nr<n)
   {
         prim=1;
         for(j=2;j<=i/2;j++)
               if(i%j==0)
                     prim=0;
               if(prim)
```

```
                {
                        int k=i,  inv=0;
                        while(k)
                        {
    inv=inv*10+k%10;
                                k=k/10;
                        }
                        for(j=2;j<=inv/2;j++)
                                if(inv%j==0)
                                        prim=0;
                                if(prim)
                                {nr++;
                                cout<<i<<" ";}
                }
                i++;
    }
                getch();
                return 1;
  }
```

Soluţie problema 16

Se parcurge numărul liniilor ce vor fi afişate, cu un contor i (i=1,n). Pentru fiecare linie parcurgem cu un contor j intervalul [1,n-i+1], lăsând spaţiile corespunzătoare fiecărei linii.

```
#include <iostream.h>
#include <conio.h>
int n,i,j;
int main()
{
   cout<<"n=";cin>>n;
   for(i=1;i<=n;i++)
   {   for(j=1;j<=i;j++)
                cout<<" ";
         for(j=1;j<=n-i+1;j++)
                cout<<j<<" ";
         cout<<endl; }
   getch();
   return 1;
}
```

Soluţie problema 17

Se citesc cele numitorul şi numărătorul fracţiei în variabilele a respectiv b. Se copiază cele două valori în variabilele a1, respectiv b1, a1=a, b1=b şi se calculează cel mai mare divizor comun dintre cele două numere. Fracţia simplificatăse obţine prin împărţirea numitorului şi numărătorului la cel mai mare divizăr comun.

```
#include <iostream.h>
int a,b,cmmdc,a1,b1;
int main()
{
   cout<<"Numitorul:";cin>>a;
   cout<<"Numaratorul:";cin>>b;
   a1=a;
   b1=b;
   while(a1!=b1)
   {
         if(a1>b1)
               a1=a1-b1;
         else
               b1=b1-a1;
   }
```

```
   cout<<"Fractia                    initiala
"<<a<<"/"<<b<<endl;

   cout<<"Fractia                simplificata
"<<a/a1<<"/"<<b/a1;

   return 1;

}
```

Soluţie problema 18

Generăm termenii şirului astfel: termenul i este format din numerele de la 1 la i, de i -1ori numărul i şi numerele de la i-1 la 1.

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

```
    getch();
    return 1;
}
```

Soluţie problema 19

Se citeşte pe rând câte un număr în variabila a, se calculează suma cifrelor şi dacă suma este divizibilă cu 3 se afişează numărul. Pentru a nu se pierde valoarea numărului citit, se reţine într-o variabila de tip întreg a1.

```
#include <iostream.h>
int n,a,i,s,a1;
int main()
{
   cout<<"n=";cin>>n;
   for(i=1;i<=n;i++)
   {
         cout<<"Numarul "<<i<<":";cin>>a;
         s=0;
         a1=a;
         while(a1!=0)
         {
               s=s+a1%10;
               a1=a1/10;
         }
```

```
            if(s%3==0)
                  cout<<a<<" ";
      }
      return 1;
}
```

Soluţie problema 20

Cât timp a este diferit de -1, se elimină câte o cifră şi se compară cu 0. În cazul în care cifra este 0, creşte valoarea variabilei nr, care la final va reţine numărul de zerouri.

```
#include <iostream.h>
int a,nr=0;
int main()
{
   cout<<"a=";cin>>a;
   while(a!=-1)
   {
         while(a!=0)
         {
               if(a%10==0)
                   nr++;
               a=a/10;
         }
         cout<<"a=";cin>>a;
   }
   cout<<"Numarul de zerouri="<<nr;
```

```
        return 1;
    }
```

Soluţie problema 21

Un număr x este pătrat perfect dacă partea întreagă din radical din x este egală cu radical din x. Se parcurg toate numerele de la 1 la n şi se verifică această proprietate.

```
#include <iostream.h>
#include <math.h>
int n,i;
int main()
{
   cout<<"n=";cin>>n;
   for(i=1;i<=n;i++)
         if(sqrt(i)==floor(sqrt(i)))
               cout<<i<<" ";
         return 1;
}
```

Soluţie problema 22

Se citesc cele două capete ale intervalului în variabilele a şi b şi se verifică validitatea intervalului. În caz afirmativ, se calculează suma divizorilor fiecărui număr din intervalul [a, b]. Dacă această sumă este egală cu i se afişează valoarea i.

```
#include <iostream.h>
int a,b,s=0,i,x;
int main()
{
   cout<<"a=";cin>>a;
   cout<<"b=";cin>>b;
   if(a>b)
         cout<<"Interval incorect";
   else
   {
         for(i=a;i<=b;i++)
         {
               x=i;s=0;
               for(int j=1;j<=x/2;j++)
                     if(x%j==0)
```

```
                        s=s+j;
                    if(s==i)
                        cout<<i<<" ";
            }
        }
        return 1;
    }
```

Soluţie problema 23

Se citesc cele n numere pe rând în variabila a. Pentru fiecare număr citit se verifică dacă este prim, şi în caz afirmativ se compară cu valoarea maximă existentă până în acel moment.

```
#include <iostream.h>
int n,max=0,a,prim,i;
int main()
{
   cout<<"n=";cin>>n;
   for(i=1;i<=n;i++)
   {
         cout<<"a=";cin>>a;
         prim=1;
         for(int j=2;j<=a/2;j++)
               if(a%j==0)
                     prim=0;
               if(prim)
                     if(a>max)
                           max=a;
   }
```

```
    cout<<"Cel mai mare numar prim="<<max;
    return 1;
}
```

Soluţie problema 24

Se împarte numărul în două părţi: restul împarţirii la 10^k , şi câtul împărţirii la 10^k .Se va forma noul număr punând pe prima poziţie restul, apoi câtul.

```
#include <iostream.h>
#include <conio.h>
#include <math.h>
long int n,c,k;
int main()
{
   cout<<"n=";cin>>n;
   cout<<"k=";cin>>k;
   c=pow(10,k);
   c=n%c;
   n=n/pow(10,k);
   long int nrcifre=0,p=n;
   while(p)
   {nrcifre++;p=p/10;}
   n=c*pow(10,nrcifre)+n;
```

```
    cout<<n;
    getch();
    return 1;
}
```

Soluţie problema 25

Căutăm intervalul în care se găsesc numerele astfel: extremitatea stângă 10^{n-1}, iar extremitatea dreaptă este formată din n cifre de 9. Se verifică fiecare număr din interval şi se aleg acele numere care au exact k cifre prime.

```
#include <iostream.h>
#include <conio.h>
int n,inv,prim,k,i,l=0,r;
int main()
{
   cout<<"n=";cin>>n;
   cout<<"k=";cin>>k;
   r=pow(10,n-1);
   for(i=1;i<=n;i++)
         l=l*10+9;
   for(i=r;i<=l;i++)
   {
         int numar=i;inv=0;
         while(numar)
         {
```

```
                inv=inv*10+numar%10;
                numar=numar/10;
        }
        if(inv==i)
        {
                int nr=0;
                numar=i;
                while(numar)
                {
    if(numar%10==2||numar%10==3
        ||numar%10==5||numar%10==7)
                                nr++;
                        numar=numar/10;
                }
                if(nr==k)
                        cout<<i<<" ";
        }
    }
    getch();
    return 1;
}
```

Soluţie problema 26

Pentru fiecare număr din intervalul [1000, 9999], se presupune că ar avea toate cifrele ordonate crescător, folosind o variabilă semafor p=1. În cazul în care p=1 se testează dacă suma cifrelor este egală cu valoarea citită s, în caz afirmativ se afişează numărul.

```
#include <iostream.h>
#include <conio.h>
int i,p,j,s1,s;
int main()
{
        cout<<"s1=";cin>>s1;
   for(i=1000;i<=9999;i++)
   {
         p=1;
         j=i; s1=0;
         while(p&&j/10)
         {
                  S1+=j%10;
               if(j%10<=(j/10)%10)
                     p=0;
```

```
                        j=j/10;
            }
            if(p&&s1==s)
                  cout<<i<<" ";
      }
      getch();
      return 1;
}
```

TABLOURI. ENUNȚURI

1. Se citesc de la tastatură un număr natural n și apoi un vector cu n elemente naturale. Afisați numărul de perechi de elemente din vectorul citit care au proprietatea că sunt prime între ele.

   Exemplu:

   Pentru n=4 și numerele 3 4 6 8 sunt 2 perechi (3,4) si (3,8).

2. Să se afişeze toate elementele dintr-un vector v care sunt prime cu un număr dat.

   Exemplu: pentru v=(12, 15, 254, 525, 56, 125, 500, 63, 48, 912) şi numărul 4, se obţin 4 numere 15, 525, 125, 63.

3. Se citeşte un vector cu n elemente numere întregi. Să se scrie un program care deplasează elementele vectorului la dreapta cu k poziţii.

   Exemplu: pentru n=6, k=3 şi vectorul v=(3, 2, 7, 10, 9, 2), după deplasare vectorul devine v=(10, 9, 2, 3, 2, 7).

4. Se citesc două numere întregi a, b. Să se scrie un program care afişează cifrele comune celor două numere o singură dată.

Exemplu: pentru a=7621 şi b=21182 se vor afişa cifrele comune 1, 2.

5. Să se scrie un program care elimină toate elemntele nule dintr-un vector de numere întregi.

Exemplu: pentru vectorul v=(14, 0, 100, 0, 0, 34), rezultă vectorul v=(14, 100, 34).

6. Se citesc două mulţimi de numere a şi b, cu n respectiv m elemente. Să se scrie un program care verifică dacă o mulţime se poate scrie ca permutarea celeilalte.

Exemplu: pentru n=4, a=(2, 4, 8, 1), m=4, b=(4, 1, 8, 2) programul va afişa mesajul ”DA”.

7. Se citesc de la tastatură cele n elemente ale unui vector. Să se afişeze toate perechile de elemente ale vectorului cu proprietatea că ambele elemente ale perechii au aceeaşi sumă a cifrelor.

Exemplu: pentru vectorul (12, 9, 54, 32, 222, 49, 34, 6, 1, 91) se vor afişa perechile (9, 54) şi (222, 6).

8. Se dă un vector cu n elemente întregi. Să se scrie un program care afişează elementul care apare de cele mai multe ori în vector.

   Exemplu: x=(15, 12, 8, 12, 19, 12, 8, 8), se vor afişa elementele 8 şi 12 care apar fiecare de 3 ori.

9. Se citeşte un vector cu n numere întregi. Să se afişeze secvenţa de lungime maximă ce se poate forma cu numere care încep cu aceeaşi cifră.

   Exemplu: pentru vectorul v=(20, 47, 81, 96, 321, 864, 8181) se va afişa secvenţa (81, 864, 8181).

10. Fie x un vector de numere întregi. Să se formeze un vector y de numere întregi, în care y[i] să fie reprezentarea în baza 2 a numărului x[i].

    Exemplu: pentru x=(12, 8, 3, 14, 6, 21), vectorul rezultat va fi y=(1100, 1000, 11, 1110, 110, 10101).

11. Se citeşte un număr natural n. Să se afişeze următorul pătrat:

A * * *.....A

* A * *..... *

* * A *..... *

.....

A *.....* A

Exemplu: pentru n=5 se afişează:

A * * * A

* A * A *

* * A * *

* A * A *

A * * * A

12. Se citeşte de la tastatură o matrice cu m linii şi n coloane de elemente numere întregi. Să se memoreze într-un vector b maximele de pe fiecare linie a matricii care sunt numere impare.

Exemplu: pentru m=4, n=5 şi matricea: $\begin{Vmatrix} 10 & 7 & 9 & 16 & 6 \\ 11 & 6 & 7 & 4 & 3 \\ 8 & 6 & 10 & 24 & 2 \\ 9 & 7 & 12 & 21 & 1 \end{Vmatrix}$

vectorul b va conţine elementele: 9, 11, 21. Nu conţine nici un element de pe linia 3 deoarece sunt numai numere pare.

13. Considerăm o matrice pătratică de dimensiune n, unde n este număr natural impar. Putem împărţi matricea în patru zone prin intermediul celor două diagonale (cea principală şi cea secundară), după cum urmează:

    a. zona 1, constituită din elementele situate deasupra diagonalei principale şi deasupra diagonalei secundare;

    b. zona 2, alcătuită din elementele situate dedesuptul diagonalei principale şi deasupra diagonalei secundare;

    c. zona 3, constituită din elementele situate dedesuptul diagonalei principale şi dedesuptul diagonalei secundare;

    d. zona 4, alcătuită din elementele situate deasupra diagonalei principale şi dedesuptul diagonalei secundare;

    Să se scrie un program care construieşte o matrice pătratică, având un număr impar n de linii şi coloane, care să conţină: valoarea 0 pentru elementele situate pe diagonale, o valoare egală cu numărul cadranului în care se află, pentru celelalte elemente.

Exemplu: pentru n=5, matricea va arăta astfel:

$$\begin{Vmatrix} 0 & 1 & 1 & 1 & 0 \\ 2 & 0 & 1 & 0 & 4 \\ 2 & 2 & 0 & 4 & 4 \\ 2 & 0 & 3 & 0 & 4 \\ 0 & 3 & 3 & 3 & 0 \end{Vmatrix}$$

14. Să se genereze o matrice pătratică de dimensiune n cu elementele 1,2,…,n^2 aşezate în zig-zag.

Exemplu: pentru o matrice de dimensiune 4:

$$\begin{Vmatrix} 1 & 2 & 3 & 4 \\ 8 & 7 & 6 & 5 \\ 9 & 10 & 11 & 12 \\ 16 & 15 & 14 & 13 \end{Vmatrix}$$

15. Să se genereze o matrice pătratică de dimensiune n cu elementele 1,2,…,n^2 aşezate în ordine crescătoare, începând cu

diagonala principală şi continuând apoi cu celelalte diagonale paralele cu aceasta. Completarea celorlalte diagonale se face alternativ, deasupra şi apoi dedesubtul diagonalei principale.

Exemplu: pentru o matrice de dimensiune 4:

$$\begin{Vmatrix} 4 & 10 & 14 & 16 \\ 5 & 3 & 9 & 13 \\ 11 & 6 & 2 & 8 \\ 15 & 12 & 7 & 1 \end{Vmatrix}$$

16. Se consideră o matrice pătratică de dimensiune n, ale cărei elemente sunt numere întregi, o parte din acestea având valoarea 0. Fără a utilize altă matrice să se mute la sfârşitul matricii toate liniile care conţin cel puţin k elemente nule, unde valoarea k se citeşte de la tastatură.

Exemplu: pentru n=4, k=2, şi matricea $\begin{Vmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 2 & 0 \\ 3 & 2 & 0 & 1 \\ 1 & 0 & 1 & 2 \end{Vmatrix}$, se va afişa matricea $\begin{Vmatrix} 3 & 2 & 0 & 1 \\ 1 & 0 & 1 & 2 \\ 2 & 0 & 0 & 0 \\ 0 & 1 & 2 & 0 \end{Vmatrix}$.

17. Se citesc două matrici patratice a şi b de dimensiuni n şi m. Să se scrie un program care calculează produsul celor două matrice.

Exemplu: pentru n=3, m=3, a= $\begin{Vmatrix} 2 & 4 & 6 \\ 10 & 4 & 7 \\ 1 & 7 & 5 \end{Vmatrix}$, b= $\begin{Vmatrix} 1 & 2 & 3 \\ 4 & 8 & 2 \\ 0 & 5 & 0 \end{Vmatrix}$, matricea

rezultat va fi: c= $\begin{Vmatrix} 18 & 66 & 14 \\ 26 & 87 & 38 \\ 29 & 83 & 17 \end{Vmatrix}$

18. Se considera o matrice A de dimensiune n*m cu elemente numere intregi. Sa se determine linia (liniile) din matrice care contine cele mai multe elemente nenule.

Exemplu : pentru n=4, m=5, a= $\begin{Vmatrix} 1 & 4 & 0 & 6 & 7 \\ 2 & 0 & 9 & 0 & 0 \\ 1 & 2 & 4 & 9 & 7 \\ 1 & 0 & 0 & 0 & 3 \end{Vmatrix}$, vor fi afişate liniile 2 şi 3, deoarece conţin cele mai multe zerouri, în număr de 3.

19. Se citeşte de la tastatură o matrice a cu m linii şi n coloane, de elemente numere întregi. Numim „marginile” matricii

cele două linii şi coloane care „încadrează” matricea. Să se scrie un program care determină cel mai mare element aflat pe marginile matricii, precum şi de câte ori apare acesta în cadrul tabloului.

Exemplu: pentru m=5, n=4 şi matricea: $\begin{Vmatrix} 3 & -4 & 8 & -2 \\ 15 & 1 & -7 & 6 \\ 3 & -7 & 15 & -8 \\ -6 & 12 & 7 & 11 \\ -1 & 2 & -5 & 7 \end{Vmatrix}$

cel mai mare element este 15, care apare de două ori în matrice.

20. Se citeşte o matrice pătratică de dimensiune n x n. Să se elimine toate coloanele care conţin cel puţin k cifre de 0, unde k se citeşte de la tastatură. Să se afişeze matricea rezultată.

Exemplu: pentru n=4, k=2 şi matricea $\begin{Vmatrix} 3 & 0 & 1 & 4 \\ 0 & 5 & 8 & 0 \\ 4 & 0 & 10 & 0 \\ 1 & 4 & 2 & 0 \end{Vmatrix}$, se va afişa matricea

$$\begin{Vmatrix} 3 & 1 \\ 0 & 8 \\ 4 & 10 \\ 1 & 2 \end{Vmatrix}.$$

21. Se citeşte o matrice pătratică de dimensiune n x n. Să se copieze elementele matricii într-un vector de dimensiune n^2, linie cu linie, să se interschimbe linia k cu linia l, şi să se reconstituie matricea modificată.

Exemplu: pentru n=4, k=2 şi l=4 şi matricea $\begin{Vmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{Vmatrix}$, va rezulta vectorul (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16) , şi matricea afişată $\begin{Vmatrix} 1 & 2 & 3 & 4 \\ 13 & 14 & 15 & 16 \\ 9 & 10 & 11 & 12 \\ 5 & 6 & 7 & 8 \end{Vmatrix}$.

Soluţie problema 1

Se citesc elementele vectorului v, apoi se parcurge vectorul și se calculează cel mai mare divizor comun între v[i] și v[j]. Dacă cel mai mare divizor comun este 1 atunci se contorizează.

```
#include <iostream>
using namespace std;
int v[1005], n;
int main ()
{
    cin >>n;
    for (int i=1; i<=n; i++)
        cin >>v[i];
    int C=0;
    for (int i=1; i<n; i++)
        for (int j=i+1; j<=n; j++)
            {
                int x=v[i], y=v[j];
                while(x!=y)
                    {
```

```
                        if(x>y)
                            x=x-y;
                        else
                            y=y-x;
                    }
                if(x==1)
                    C++;
            }
    cout <<C;
    return 0;
}
```

Soluţie problema 2

Pentru fiecare număr din vector se presupune că ar fi prim cu numărul citit. Dacă se găseşte un divizor al numărului v[i] diferit de 1, se verifică dacă este şi divizor al numărului citit.

```
#include <iostream.h>
int v[20],n,k,i;
int main()
{
      cout<<"n=";cin>>n;
      for(i=1;i<=n;i++)
      {cout<<"v["<<i<<"]=";cin>>v[i];}
      cout<<"k=";cin>>k;
      for(i=1;i<=n;i++)
      {   int prim=1;
            for(int j=2;j<=v[i]/2;j++)
                  if(v[i]%j==0        &&
k%j==0)
                        prim=0;
                  if(prim)
                        cout<<v[i]<<"
";
```

```
            }
            return 1;
        }
```

Soluţie problema 3

Se citesc cele n elemente ale vectorului v şi numărul de deplasări în variabila k. Se vor parcurge următorii paşi:

- pentru i=1, n se execută instrucţiunile:
- se reţine de fiecare dată în variabila x ultimul element din vector;
- se deplasează elementele vectorului cu o poziţie la dreapta prin copierea elementelor de pe poziţiile j, pe poziţiile j+1, j=n-1,1;
- se inserează valoarea variabilei x pe prima poziţie în vector.

```
#include <iostream.h>
int n,v[100],i,k,j,x;
int main()
{
   cout<<"n=";cin>>n;
   for(i=1;i<=n;i++)
   {
         cout<<"v["<<i<<"]=";cin>>v[i];
   }
   cout<<"k=";cin>>k;
```

```
    for(i=1;i<=k;i++)
    {
          x=v[n];
          for(j=n-1;j>=1;j--)
                v[j+1]=v[j];
          v[1]=x;
    }
    for(i=1;i<=n;i++)
          cout<<v[i]<<" ";
    return 1;
}
```

Soluţie problema 4

Se utilizează doi vectori va şi vb cu 10 elemente, în care vom reţine pe poziţia i valoarea 1 dacă cifra i există în numărul a (pentru vectorul va) şi b (pentru vectorul vb), şi valoarea 0 în caz contrar (i=0, 9). Se compară cei doi vectori şi pe poziţiile comune care conţin valoarea 1, se afişează acea poziţie care reprezintă o cifra comună.

```
#include <iostream.h>
int a, b, va[10], vb[10],i,j;
int main()
{
   cout<<"Primul numar=";cin>>a;
   cout<<"Al doilea numar=";cin>>b;
   for(i=0;i<=9;i++)
         va[i]=vb[i]=0;
   while(a)
   {
         va[a%10]=1;
         a=a/10;
   }
   while(b)
```

```
    {
          vb[b%10]=1;
          b=b/10;
    }
    cout<<"Cifrele comune:\n";
    for(i=0;i<=9;i++)
          if((va[i]!=0)&&(va[i]==vb[i]))
                cout<<i<<" ";
    return 1;
}
```

Soluţie problema 5

Parcurgem vectorul şi căutăm prima valoare de zero. Eliminăm zeroul de pe poziţia i, prin deplasarea tuturor elementelor de pe poziţiile i+1, i+2,…..,n cu o poziţie la stânga. Vom folosi o structură repetitivă de tip while, while(v[i]==0), pentru a elimina (dacă există), zerourile aflate pe poziţii consecutive. După fiecare element eliminat se scade numărul de elemente, n--.

```
#include <iostream.h>
int v[20],n,i,j;
int main()
{
      cout<<"n=";cin>>n;
      for(i=1;i<=n;i++)
      { cout<<"v["<<i<<"]=";cin>>v[i];}
      for(i=1;i<=n;i++)
      {
            while(v[i]==0)
            {
                  for(j=i;j<n;j++)
                        v[j]=v[j+1];
                  n--;
```

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

Soluţie problema 6

Se verifica mai intai dacă numărul de elemente al primei mulţimi este egal cu numărul de elemente al celei de-a doua mulţimi.Dacă sunt diferite o multime nu poate fi scrisă ca permutarea celeilalte, in caz contrar se ordonează crescător cele două mulţimi şi se verifică dacă sunt egale.

```
#include <iostream.h>

int a[50], b[50],n,m,i,j;

int main()

{

   cout<<"Introduceti nr elemente prima
multime=";

   cin>>n;

   cout<<"Introduceti nr elemente a doua
multime=";

   cin>>m;

   cout<<"Introduceti prima multime:\n";

   if(n!=m)

      cout<<"O multime nu poate fi
permutarea celeilalte";

   else

   {
```

```
    for(i=1;i<=n;i++)
          cin>>a[i];
    cout<<"Introduceti a doua multime:\n";
    for(i=1;i<=m;i++)
          cin>>b[i];
                //ordonam  crescator  prima
multime
          for(i=1;i<n;i++)
                for(j=i+1;j<=n;j++)
                      if(a[i]>a[j])
                      {
                            int aux=a[i];
                            a[i]=a[j];
                            a[j]=aux;
                      }
          //odonam  crescator  cea  de-a  doua
multime
          for(i=1;i<n;i++)
                for(j=i+1;j<=n;j++)
                            if(b[i]>b[j])
                            {
```

```
                                        int
aux=b[i];

   b[i]=b[j];

                                        b[j]=aux;

                                    }

                                    int p=1;

   for(i=1;i<=n;i++)

   if(a[i]!=b[i])

   p=0;

                                        if(p)

   cout<<"DA";

                                        else

   cout<<"NU";

   }

   return 1;

}
```

Soluţie problema 7

Pentru fiecare element v[i] , calculăm suma cifrelor sale şi căutăm perechea printre elementele v[j], cu j=1,n. Când s-a găsit o pereche de elemnte (v[i], v[j]) pentru care suma cifrelor elementului v[i] este egală cu suma cifrelor elementului v[j], o afişăm pe ecran.

```
#include <iostream.h>
int v[20],n,i,j,s,s1,k,k1;
int main()
{
       cout<<"n=";cin>>n;
       for(i=1;i<=n;i++)
       {
              cout<<"v["<<i<<"]=";cin>>v[
       i];
       }
       for(i=1;i<n;i++)
       {
              k=v[i];s=0;
              while(k)
              {
                     s+=k%10;
```

```
                    k=k/10;
                }
                for(j=i+1;j<=n;j++)
                {
                k1=v[j];s1=0;
                      while(k1)
                      {
                            s1+=k1%10;
                            k1=k1/10;
                      }
                      if(s==s1)
cout<<"("<<v[i]<<","<<v[j]<<")"<<"   ";
                }
      }
      return 1;
}
```

Soluţie problema 8

Vom folosi un vector nrap, în care vom reţine pe fiecare poziţie i, câte elemente de pe poziţiile i+1, i+2, …., n sunt egale cu v[i]. În vectorul nrap, vom avea valoare maximă pe poziţiile pe care se găsesc elementele din vectorul v care apar de cele mai multe ori. Se parcurge vectorul nrap şi de pe poziţiile i, unde nrap[i]=max, se afişează elementul v[i].

```
#include <iostream.h>
int v[20],n,i,nrap[20],max=0,j;
int main()
{
	cout<<"n=";cin>>n;
	for(i=1;i<=n;i++)
	{
		cout<<"v["<<i<<"]=";
		cin>>v[i];
	}
	for(i=1;i<=n;i++)
		nrap[i]=1;
	for(i=1;i<n;i++)
		for(j=i+1;j<=n;j++)
```

```
                    if(v[i]==v[j])
                          nrap[i]++;
                    for(i=1;i<=n;i++)
                          if(nrap[i]>max)

      max=nrap[i];

      for(i=1;i<=n;i++)

      if(max==nrap[i])
      cout<<v[i]<<"   "<<"de   "<<max<<"
ori"<<endl;
                                  return 1;
}
```

Soluţie problema 9

Pentru fiecare număr v[i] vom reţine într-un vector a pe poziţia i prima cifră a numărului v[i]. Vom folosi un vector cif, care să conţină numărul de apariţii al fiecărei cifre în vectorul a, după care vom reţine în variabila max valoarea maximă din vectorul cif. Parcurgem vectorul cif şi căutăm valoarea max, iar poziţia pe care se găseşte valoarea max reprezintă cifra din vectorul a, a cărei poziţie din vectorul a, reprezintă poziţia corespunzătoare numărului ce trebuie afişat din vectorul v.

```
#include <iostream.h>

int
v[50],cif[50],a[50],max,i,j,n,nr;

int main()

{

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

   for(i=1;i<=n;i++)

         {


   cout<<"v["<<i<<"]=";cin>>v[i];

         }

   for(i=1;i<=n;i++)

   {

         nr=0;
```

```
        int x=v[i];
        while(x)
        {
                x=x/10;
                nr++;
        }
        a[i]=v[i]/pow(10,nr-1);
}
for(i=0;i<=9;i++)
        cif[i]=0;
for(i=1;i<=n;i++)
        cif[a[i]]++;
max=0;
for(i=0;i<=9;i++)
        if(cif[i]>max)
                max=cif[i];
        for(i=0;i<=9;i++)
                if(cif[i]==max)
                {

for(j=1;j<=n;j++)
```

```
    if(a[j]==i)

    cout<<v[j]<<" ";

    cout<<endl;
                }
    return 1;
}
```

Soluţie problema 10

Fiecare număr din vectorul x îl vom reţine în variabila k care va fi prelucrat şi transformat în baza 2. Pentru transformarea numărului k în baza 2 vom folosi următorul algoritm: împărţim numărul la doi cât timp câtul este diferit de 1, şi restul îl adăugam de fiecare dată la sfârşitul numărului transformat. În final vom adăuga şi câtul rămas pe prima poziţie la numărul transformat. Numărul în baza 2 va fi reţinut în variabila nr, care apoi va fi copiat în vectorul y pe poziţia i.

```
#include <iostream.h>
long int x[50],y[50],k,p,nr;
int i,j,n;
int main()
{
   cout<<"n=";cin>>n;
   for(i=1;i<=n;i++)
   {
          cout<<"x["<<i<<"]=";
          cin>>x[i];
   }
   for(i=1;i<=n;i++)
   {
```

```
            k=x[i];
            p=1;
            nr=0;
            while(k!=1)
            {
                  nr= nr+(k%2)*p;
                  p=p*10;
                  k=k/2;
            }
            nr=nr+k*p;
            y[i]=nr;
      }
      for(i=1;i<=n;i++)
            cout<<y[i]<<" ";
      return 1;
}
```

Soluţie problema 11

Se iniţializează matricea cu caracterul ‘ * ‘, după care se parcurg cele două diagonale şi sunt completate cu caracterul ‘ A ‘. Pentru cele două diagonale se parcurg valorile de la 1 la n cu un indice i şi a[i][i]='A', a[i][n-i+1]='A'. În final se afişează elementele matricei.

```
#include <iostream.h>
int i,j,n;
int main()
{
   cout<<"n=";cin>>n;
   for(i=1;i<=n;i++)
         for(j=1;j<=n;j++)
               a[i][j]='*';
         for(i=1;i<=n;i++)
         {
               a[i][i]='A';
               a[i][n-i+1]='A';
         }
         for(i=1;i<=n;i++)
         {
```

```
            for(j=1;j<=n;j++)
                  cout<<a[i][j]<<" ";
            cout<<endl;
      }
      return 1;
}
```

Soluţie problema 12

Parcurgem matricea linie cu linie şi pentru fiecare linie căutăm valoarea maximă, ce va fi reţinută în variabila max. Testăm dacă maximul este impar şi în caz afirmativ îl reţinem în vectorul v.

```
#include <iostream.h>
int a[50][50],n,m,i,j,max,v[50],k;
int main()
{
   cout<<"m=";cin>>m;
   cout<<"n=";cin>>n;
   for(i=1;i<=m;i++)
         for(j=1;j<=n;j++)
         {
   cout<<"a["<<i<<"]["<<j<<"]=";
               cin>>a[i][j];
         }
         for(i=1;i<=m;i++)
         {
               max=a[i][1];
```

```
                for(j=1;j<=n;j++)
                        if(a[i][j]>max)
                                max=a[i][j];
                        if(max%2!=0)
                                v[++k]=max;
        }

   for(i=1;i<=k;i++)
                                cout<<v[i]<<"
";
                                return 1;
}
```

Soluţie problema 13

Iniţializăm matricea cu 0, apoi pe rând generăm elementele celor 4 triunghiuri bazându- ne pe proprietatea de simetrie faţă de linii şi coloane. Triunghiul format în partea de sus a diagonalei principale şi secundare este simetric cu triunghiul format de partea de jos a diagonalei principale şi diagonalei secundare, iar triunghiul din stânga format de partea de sus a diagonalei principale şi partea de jos a diagonalei secundare este simetric cu triunghiul din dreapta format de partea de sus a diagonalei secundare şi partea de jos a diagonalei principale.

```
#include <iostream.h>
int a[20][20],n,i,j,k=1;
int main()
{
   cout<<"n=";cin>>n;
   for(i=1;i<=n;i++)
         for(j=1;j<=n;j++)
               a[i][j]=0;
         for(i=1;i<=n/2;i++)
              for(j=i+1;j<=n-i;j++)
                {
   a[i][j]=k;//pentru triunghiul de sus
```

```
   a[n-i+1][j]=k+2;// triunghiul de jos
   a[j][i]=k+1;//triunghiul din stanga
   a[j][n-i+1]=k+3;//triunghiul         din
dreapta
                  }
                    for(i=1;i<=n;i++)
               {     for(j=1;j<=n;j++)
                     cout<<a[i][j]<<" ";
                     cout<<endl;
               }
               return 1;
   }
```

Soluţie problema 14

Vom parcurge liniile matricii a de la 1 la n într-un ciclu while, şi vom genera pe rând într-un ciclu for elementele de pe liniile impare de la 1 la n, şi într-un alt ciclu for elementele de la n la 1. Vom folosi o variabilă k, iniţializată la început cu valoarea 0, pentru a reţine pe rând fiecare din cele n^2 numere.

```
#include <iostream.h>
int a[50][50],n,i,j,k=0,l;
int main()
{
   cout<<"n=";cin>>n;
   i=1;
    while(i<=n)
   {
         for(j=1;j<=n;j++)
               a[i][j]=++k;
              i++;
         for(j=n;j>=1;j--)
           a[i][j]=++k;
         i++;
```

```
    }
    for(i=1;i<=n;i++)
    {
          for(j=1;j<=n;j++)
                cout<<a[i][j]<<" ";
          cout<<endl;
    }
    return 1;
}
```

Soluţie problema 15

Generăm într – o primă etapă elementele de pe diagonala principală de jos în sus după formula a[n-i+1][n-i+1]. Parcurgem cu un indice i numărul de diagonale rămase de generat (i=1,n-1), diagonale ce sunt paralele cu diagonala principală.

```
#include <iostream.h>
int a[20][20],n,i,j,k=0;
int main()
{
   cout<<"n=";cin>>n;
   for(i=1;i<=n;i++)
         a[n-i+1][n-i+1]=++k;
   for(i=1;i<n;i++)
   {

         for(j=i+1;j<=n;j++)
               a[j][j-i]=++k;
         for(j=n-i;j>=1;j--)
            a[j][j+i]=++k;
   }
```

```
    for(i=1;i<=n;i++)
    {
          for(j=1;j<=n;j++)
                cout<<a[i][j]<<" ";
          cout<<endl;
    }
    return 1;
}
```

Soluţie problema 16

Vom parcurge liniile matricii de la n la 1, cu un indice i, pentru fiecare linie i numărăm în variabila nr câte valori de 0 conţine. Dacă numărul de valori de 0 este mai mare sau egal decât k, copiem linia i într-un vector v şi deplasăm liniile q (q=i+1,l), cu o poziţie în sus.Variabila l reţine numărul liniei de jos în sus unde vom copia elementele liniei i reţinute în vectorul v. Iniţial l=n+1.

```
#include <iostream.h>
int a[50][50],n,i,j,k,l,v[50],q;
int main()
{
   cout<<"n=";cin>>n;
   for(i=1;i<=n;i++)
         for(j=1;j<=n;j++)
         {
cout<<"a["<<i<<"]["<<j<<"]=";cin>>a[i][j];
}
         cout<<"k=";cin>>k;
         for(i=1;i<=n;i++)
         {
               for(j=1;j<=n;j++)
                     cout<<a[i][j]<<" ";
```

```
                cout<<endl;
        }
        int nr=0;
        l=n+1;
                for(i=n;i>=1;i--)
                {
                        nr=0;
                        for(j=1;j<=n;j++)
                                if(a[i][j]==0)
                                        nr++;
for(q=1;q<=n;q++)
                                v[q]=0;
                                if(nr>=k)
                                {       l--;
for(j=1;j<=n;j++)
                                  v[j]=a[i][j];
                        for(q=i;q<=l;q++)
                        for(j=1;j<=n;j++)
                                a[q][j]=a[q+1][
                        j];
                                for(j=1;j<=n;j+
```

```
                        +)
                                a[l][j]=v[j];
                                }
                }
                cout<<endl;
                for(i=1;i<=n;i++)
                {
                        for(j=1;j<=n;j++)
                        cout<<a[i][j]<<" ";
                        cout<<endl;
                }
    return 1;
}
```

Soluţie problema 17

Se citesc cele două matrice a şi b, unde numărul de coloane al primei matrice să coincidă cu numărul de linii al celei de-a doua matrice, pentru a se putea realiza operaţia de înmulţire. Conform algoritmului matematic un element al matricei produs $c_{ij} = \sum_{i,j,k=1}^{n} a_{ik} \bullet b_{kj}$. Se afişează matricea produs c.

```
#include <iostream.h>

int main()

{

 int
n,m,p,i,j,k,a[10][10],b[10][10],c[10][10];

 cout<<"Dati      dimensiunile      matricei
"<<endl;

 cout<<"Dati  numarul  de  linii      n  =
";cin>>n;

 cout<<"Dati  numarul  de  coloane  m  =
";cin>>m;

 for(i=1;i<=n;i++)

  for(j=1;j<=m;j++)

{

   cout<<"a["<<i<<","<<j<<"]= ";
```

```
                        cin>>a[i][j];
}
 cout<<"Elementele matricei A sunt:
"<<endl;
 for(i=1;i<=n;i++)
{
  for(j=1;j<=m;j++)
   cout<<a[i][j]<<" ";
  cout<<endl;
}
 cout<<"Dati numarul de linii      m =
";cin>>m;
 cout<<"Dati numarul de coloane p =
";cin>>p;
 for(i=1;i<=m;i++)
  for(j=1;j<=p;j++)
{
   cout<<"b["<<i<<","<<j<<"]= ";
                        cin>>b[i][j];
}
 cout<<"Elementele matricei b sunt:
"<<endl;
 for(i=1;i<=m;i++)
```

```
{
  for(j=1;j<=p;j++)
   cout<<b[i][j]<<" ";
  cout<<endl;
}
 for(i=1;i<=n;i++)
  for(j=1;j<=p;j++)
   for(k=1;k<=m;k++)
    c[i][j]+=a[i][k]*b[k][j];
        cout<<"Elementele  matricei  produs
"<<endl;
 for(i=1;i<=n;i++)
{
  for(j=1;j<=p;j++)
   cout<<c[i][j]<<" ";
  cout<<endl;
}
}
```

Soluţie problema 18

Se citesc cele n*m elemente ale matricei a, apoi se parcure matricea linie cu linie, determinându-se numărul de zerouri existente pe fiecare linie. Vom utiliza o variabilă max, care iniţial va fi 0, şi după ce s-a determinat numărul de zerouri de pe linia i, se compară cu max. Dacă acesta este mai mare ca max, atunci noul max va fi acel număr.

```
#include<iostream.h>

int main()

{

 int a[30][30],n,m,i,j,max,nr;

 cout<<"Dati numarul de linii      n =
";cin>>n;

 cout<<"Dati numarul de coloane m =
";cin>>m;

 for(i=1;i<=n;i++)

  for(j=1;j<=m;j++)

{

    cout<<"a["<<i<<","<<j<<"]= ";

    cin>>a[i][j];

}
```

```
 cout<<endl<<"Matricea         A         are
elementele:"<<endl;

 for(i=1;i<=n;i++)

   {

  for(j=1;j<=m;j++) cout<<a[i][j]<<" ";

  cout<<endl;

   }

 max=0;

 for(i=1;i<=n;i++)

   {

  nr=0;

  for(j=1;j<=m;j++)

   if(a[i][j]!=0) nr++;

  if(max<nr) max=nr;

   }

 for(i=1;i<=n;i++)

   {

  nr=0;

  for(j=1;j<=m;j++)

   if(a[i][j]!=0) nr++;

   if(max==nr)
```

```
    {
    cout<<"Linia   "<<i<<"   are   "<<max<<"
elemente nenule";
    cout<<endl;
    }
}
}
```

Soluţie problema 19

Parcurgem conturul matricii şi în acelaşi timp căutăm valoarea maximă ce va fi reţinută în variabila max. Printr- o parcurgere cu i=1,n căutăm maximul pe linia de sus şi cea de jos, şi printr-o parcurgere cu i=2,m vom găsi maximul pe coloane. Parcurgem apoi matricea şi vom reţine în variabila nr câte elemente au valoarea egală cu max.

```
#include <iostream.h>
int a[50][50],n,i,j,m,max=0,nr=0;
int main()
{
   cout<<"m=";cin>>m;
   cout<<"n=";cin>>n;
   for(i=1;i<=m;i++)
         for(j=1;j<=n;j++)
         {
         cout<<"a["<<i<<"]["<<j<<"]=";
               cin>>a[i][j];
         }
         for(i=1;i<=n;i++)
               {
```

```
                        if(a[1][i]>max)
                        max=a[1][i];
                        if(a[m][i]>max)
                              max=a[m][i];
                  }
            for(i=2;i<m;i++)
            {
                  if(a[i][1]>max)
                        max=a[i][1];
                  if(a[i][n]>max)
                        max=a[i][n];
            }
            for(i=1;i<=m;i++)
                  for(j=1;j<=n;j++)
                        if(a[i][j]==max)
                              nr++;
            cout<<"Valoarea maxima:"<<max;
            cout<<" apare de "<<nr<<" ori";
                        return 1;
}
```

Soluţie problema 20

Parcurgem matricea pe coloane şi pentru fiecare coloană numărăm în variabila nr câte elemente de 0 avem. Dacă numărul de zerouri găsite este mai mare sau egal cu valoarea k, atunci coloana i va fi eliminată. Eliminarea coloanei i se va realiza prin copierea coloanelor i+1, i+2,….m, cu o poziţie la stânga, şi la fiecare eliminare de coloană scădem valoarea m cu o unitate. Variabila m am utilizat-o pentru a reţine numărul de coloane din matricea a. Am reţinut de la început valoarea n, deoarece numărul de coloane se va modifica pe parcurs. Am folosit un ciclu repetitiv while pentru a putea elimina eventualele coloane consecutive ce au număr de zerouri mai mare sau egal cu valoarea k.

```
#include <iostream.h>
 int a[50][50],n,i,j,p,q,m,nr=0,k;
 int main()
 {
 clrscr();
 cout<<"n=";cin>>n;
       for(i=1;i<=n;i++)
             for(j=1;j<=n;j++)
             {
       cout<<"a["<<i<<"]["<<j<<"]=";
       cin>>a[i][j];
```

```
                }
            cout<<endl;
            m=n;
            for(i=1;i<=n;i++)
            {
                for(j=1;j<=m;j++)
                cout<<a[i][j]<<" ";
                cout<<endl;
            }
     cout<<"k=";cin>>k;
   for(i=1;i<=m;i++)
   {    nr=0;
             for(j=1;j<=n;j++)
     if(a[j][i]==0)
                  nr++;
       while(nr>=k&& i!=m)
     {
            for(p=i+1;p<=m;p++)
              for(q=1;q<=n;q++)
            a[q][p-1]=a[q][p];
        m--;
```

```
      nr=0;
      for(j=1;j<=n;j++)
       if(a[j][i]==0)
       nr++;
   }
   if(i==m&&nr>=k)
   m--;}
    for(i=1;i<=n;i++)
    {
          for(j=1;j<=m;j++)
          cout<<a[i][j]<<" ";
          cout<<endl;}
return 1;
}
```

Soluţie problema 21

Pentru început vom copia elementele matricii a într-un vector de dimensiune n*n linie cu linie.Vom interschimba între ele liniile k şi l reţinute în vector astfel: parcurgem cu un indice i linia k, (i=n*k-n+1,n*k) şi schimbăm elementele cu cele de pe linia l , v[n*l-n+p]. n*k-n+1, reprezintă elementul de început al liniei k, unde n*k este ultimul element poziţionat pe linia k, n*k-n reprezintă ultimul element de pe linia k-1(dacă k nu este prima linie), n*k-n+1 reprezintă primul element de pe linia k. Copiem elementele vectorului v în matricea a, linie cu linie şi afişăm matricea.

```
#include <iostream.h>

int a[50][50],n,i,j,k,l, v[100],p=0;

int main()

{

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

   for(i=1;i<=n;i++)

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

         for(i=1;i<=n;i++)

               for(j=1;j<=n;j++)

                     v[++p]=a[i][j];

               for(i=1;i<=n;i++)
```

```
                    {
                          for(j=1;j<=n;j++)
                          cout<<a[i][j]<<"
";cout<<endl;}
                    cout<<"k=";cin>>k;
                    cout<<"l=";cin>>l;
                    p=1;
                    for(i=n*k-n+1;i<=n*k;i++)
                    {
                          int aux=v[i];
                          v[i]=v[n*l-n+p];
                          v[n*l-n+p]=aux;
                          p++;
                    }
                    p=0;
                    for(i=1;i<=n;i++)
                          for(j=1;j<=n;j++)
                                a[i][j]=v[++p];
                          cout<<endl;
                          for(i=1;i<=n;i++)
                          {
```

```
                    for(j=1;j<=n;j++)
                      cout<<a[i][j]<<"
";
                        cout<<endl;}
                   return 1;}
```

FIȘIERE. ENUNȚURI

1. Se consideră fisierul NUMERE.IN ce conține un șir crescător cu cel mult un milion de numere naturale de cel mult nouă cifre fiecare, separate prin câte un spatiu. Să se scrie un program C/C++ care, folosind un algoritm eficient din punct de vedere al memoriei utilizate și al timpului de executare, citește din fișier toți termenii sirului și afișează pe ecran, pe o singură linie, fiecare termen distinct al șirului urmat de numărul de apariții ale acestuia în șir. Valorile afișate sunt separate prin câte un spatiu.

   Exemplu:

NUMERE. IN	se afișează
1 1 1 5 5 5 5 9 9 11 20 20 20 1 20 3	1 3 5 4 9 2 11 1 20 1

2. Se citesc din fişierul “NOTE.IN”, de pe prima linie numărul n de note, iar de pe următoarele n linii câte trei note de la 1 la 10, numere reale. Să se afişeze în fişierul “MEDII.OUT”, mediile aritmetice pentru fiecare linie.

   Exemplu:

NOTE.IN	MEDII.OUT
5	

10 4.50 9

9 5 7.40

3.50 6 7.25

10 10 10

7.50 6.50 8

3. Se citeşte un şir de numere întregi din fişierul ‘impare.in’. Fişierul conţine: pe primul rând numărul n de elemente ale şirului separate prin spaţii. Să se afişeze în fişierul ’impare.out’, elementele impare din şir.

Exemplu:

impare.in	impare.out
7	5 -9 7 43
12 -4 5 -9 10 7 43	

4. Fişierul text ‘numere.txt‘, conţine mai multe rânduri de numere, pozitive şi negative, valorile de pe fiecare linie fiind separate prin spaţii, iar şirul acestora se încheie cu valoarea 0. Să se scrie un program care afişează pe ecran rândul(rândurile) de lungime maximă.

Exemplu:

numere.txt

-2 4 6 10 0

11 7 9 8 -9 0

1 18 0

vor fi afişate valorile : 4 5 2, si rândul pe care se află şirul de lungime maximă 2.

5. Să se scrie un program care citeşte din fişierul ‘romb.in‘, o valoare întreagă n, şi tipăreşte în fişierul ‘romb.out‘, un romb compus din n linii de caractere ‘X‘, fiecare rand conţinând caracterul ‘X‘ multiplicat astfel: pe primul rand o dată, pe al doilea rand de trei ori, pe al treilea rand de cinci ori, pe al patrulea rand de 7 ori, etc, până la jumătatea rombului, apoi numărul de caractere de pe fiecare rand descreşte în sens invers după acelaşi algoritm. Dacă n are valoare pară atunci acesta se măreşte cu 1.

Exemplu:

romb.in	romb.out
7	X
	XXX
	XXXXX
	XXXXXXX
	XXXXX

XXX

X

6. Se citeşte un şir de numere întregi din fişierul text ‘numere.in‘. Toate elementele şirului sunt scrise în fişier pe un singur rând, separate prin spaţii şi nu se cunoaşte numărul elementelor. Să se afişeze în fişierul ‘numere.out‘, elementele distincte ale şirului pe două rânduri astfel: pe primul rând se vor scrie elementele pare în ordine crescătoare, iar pe al doilea rând elementele impare tot în ordine crescătoare.

   Exemplu:

   numere.in — numere.out

   12 4 7 12 3 9 10 11 7 4 6 — 4 6 10 12

   3 7 9 11

7. Din fişierul ’numere.in’ se citesc, de pe prima linie un număr n întreg iar de pe următoarea linie n numere întregi. Se cere să se creeze două fişiere ’pare.out’, ’impare.out’ care să conţină numerele pare respectiv impare din fişierul de intrare.

   Exemplu:

numere.in pare.out
impare.out

8 4 12 8 10 3 67 1
17

4 12 3 67 8 10 1 17

8. Se citeşte o matrice pătrată de dimensiune n din fişierul ‘matrice.in‘.Pe prima linie a fişierului se află dimensiunea n, iar pe fiecare dintre următoarele n rânduri elementele fiecărei linii separate prin spaţii. Să se scrie un program care interschimbă două coloane k, l, afişând rezultatul în fişierul ‘matrice.out‘.

Exemplu: pentru k=2, l=3:

matrice.in	matrice.out
4 2 3	10 6 4 1
10 4 6 1	9 4 12 12
9 12 4 12	4 2 9 8
4 9 2 8	1 3 2 4
1 2 3 4	

9. Din fişierul “numere.in” se citesc : de pe prima linie un număr natural n, şi de pe următoarea linie n numere întregi separate prin câte un spaţiu. Să se scrie un program

care afişează în fişierul “numere.out”, toate numerele din primul fişier care au cel mult două cifre.

Exemplu:

numere.in	numere.out
7	10 1 12
102 10 1 150 3514 12 419	

10. Fişierul “numere.in” conţine pe o singură linie separate prin câte un spaţiu, cel mult 100 numere întregi, de maxim 4 cifre. Să se scrie un program care citeşte numerele din fişierul “numere.in” şi afişează în fişierul “numere.out” separate prin câte un spaţiu şi ordonate crescător toate elementele poziţive. În cazul în care nu există nici un număr pozitiv se va afişa mesajul “nu există numere pozitive”.

Exemplu:

numere.in	numere.out
20 -14 6 -9 25 -100	6 20 25

11. Se citeşte un număr întreg din fişierul text ’divizori.in’. Să se determine toate perechile de numere naturale mai mici

sau egale decât n, cu proprietatea că cel mai mare divizor comun al elementelor perechii este o valoare dată d. Fiecare astfel de pereche va fi scrisă pe un rând al fişierului 'divizori.out', cu numerele ce o compun separate prin spaţii.

Exemplu:

divizori.in	divizori.out
6 2	2 4
	2 6
	4 6

12. Se consideră fişierul 'fracţii.in' care conţine pe fiecare linie câte două numere naturale mai mici decât 35000, reprezentând numărătorul, respectiv numitorul unei fracţii simple. Să se creeze fişierul 'fracţii.out', care va conţine pe fiecare rând numărătorii şi numitorii fracţiilor ireductibile rezultate prin simplificarea fracţiilor din fişierul 'fracţii.in'.

Exemplu:

fracţii.in	fracţii.out
10 12	5 6
15 25	3 5
50 20	5 2

13. dsadsad Fişierul text ‘litera.in‘ conţine pe prima linie un număr întreg n (1<=n<=24). Pe următoarele n linii se găsesc câte o literă c urmată de un întreg n separate prin câte un spaţiu. Se cere să se construiască un fişier ‘litera.out‘ care să conţină pe fiecare linie caracterul c repetat de n ori..

Exemplu:

litera.in	litera.out
6	bbbbb
b 5	xxxx
x 4	llllll
l 6	uu
u 2	tt
t 3	kkkk
k 4	

14. Se citeşte de la tastatură un număr natural n. Să se scrie un program care scrie în fişierul ’triunghi.txt’ un triunghi de numere ca în exemplul de mai jos.

Exemplu: pentru n=7

triunghi.txt

1

1 3

1 3 5

1 3 5 7

1 3 5 7 9

1 3 5 7 9 11

1 3 5 7 9 11 13

15. Se consideră fişierul “sir.in” ce conţine un şir crescător de cel mult 1000 de numere întregi, de cel mult opt cifre fiecare separate de câte un spaţiu. Să se scrie un program care citeşte din fişier toţi termenii şirului şi afişează în fişierul ”sir.out”, pe o singură linie, fiecare termen distinct al şirului o singură dată.

Exemplu:

sir.in	sir.out
2 2 4 4 4 10 10 18 18 18	2 4 10 18

16. Pe prima linie a fişierului ”numere.in” se găsesc, separate prin câte un spaţiu, mai multe numere naturale formate

din maxim 9 cifre fiecare. Să se scrie un program care citeşte toate numerele din fişier, elimină toate cifrele pare din fiecare dintre aceste numere şi apoi scrie în fişierul text ”numere.out”, numerele obţinute.

Exemplu:

numere.in
numere.out

28 17 90 8102 3147 98 773 17 9 1 317 9 773

17. Se citeşte din fişierul ‘matrice.in‘ o matrice pătrată de dimensiune n, unde pe prima linie a fişierului se află valoarea n, şi pe fiecare din cele n rânduri se află câte o linie a matricii, elemente separate prin spaţiu. Să se scrie în fişierul ‘matrice.out‘ elementele situate sub diagonala principală(inclusiv cele de pe diagonală).

Exemplu:

matrice.in

4

1 2 3 4

5 6 7 8

9 10 11 12

matrice.out

1

5 6

9 10 11

13 14 15 16 13
14 15 16

18. În fişierul “numere.in” sunt memorate maxim 1000 numere naturale de maxim 9 cifre fiecare, toate pe o singură linie. Să se afişeze în fişierul ”numere.out” în ordine descrescătoare toate cifrele care apar în numerele din fişier.

Exemplu:

numere.in
numere.out

476
9887654322

2895

3482

19. Fişierul numar.in conţine pe prima linie un număr natural n iar pe următoarea linie n numere de cel mult 9 cifre separate prin câte un spaţiu. Să se afişeze în fişierul numar.out toate numerele formate din cifre distincte.

Exemplu:

numar.in
numar.out

122

6 176 8792 9817
109

1231 176 8792 9817 2424 109

20. Fetiţele din grupa mare de la grădiniţă culeg flori şi vor să împletească coroniţe pentru festivitatea de premiere. În grădină sunt mai multe tipuri de flori. Fiecare dintre cele `n` fetiţe culege un buchet având acelaşi număr de flori, însă nu neapărat de acelaşi tip. Pentru a împleti coroniţele fetiţele se împart în grupe. O fetiţă se poate ataşa unui grup numai dacă are cel puţin o floare de acelaşi tip cu cel puţin o altă fetiţă din grupul respectiv.

(Olimpiada de Informatică – faza judeţeană 19 martie 2006)

Cerinţă

Fiind dat un număr natural **`n`** reprezentând numărul fetiţelor şi numărul natural **`k`** reprezentând numărul de flori dintr-un buchet, să se determine grupele care se formează.

Date de intrare

Fişierul de intrare **`flori.in`** conţine pe prima linie, separate printr-un spaţiu, numerele naturale **`n`** şi **`k`**, reprezentând numărul de fetiţe şi respectiv numărul de flori din fiecare buchet. Fiecare dintre următoarele **`n`** linii

conţine, pentru fiecare fetiţă, câte **k** valori separate prin câte un spaţiu reprezentând tipurile de flori culese.

Date de ieşire

Fişierul de ieşire **flori.out** va conţine pe fiecare linie câte o grupă formată din numerele de ordine ale fetiţelor separate prin câte un spaţiu, în ordine crescătoare, ca în exemplu.

Restricţii şi precizări

- `1<=n<=150`
- `1<=k<=100`
- Tipul unei flori este un număr întreg din intervalul [`0,100`].
- Într-o grupă numerele de ordine ale fetiţelor trebuie date în ordine strict crescătoare.
- În fişierul de ieşire grupele vor fi afişate în ordinea crescătoare a numărului de ordine al primei fetiţe din grupă.

flori.in	**flori.out**	**Explicaţii**

5 4 **1 2 3 4** **5 6 9 6** **1 1 1 1** **2 4 4 3** **7 7 7 7**	**1 3 4** **2** **5**	Fetiţele **1** şi **3** au cules amândouă flori de tipul 1, iar fetiţele **1** şi **4** au cules amândouă flori de tipurile 2,3 şi 4, deci toate cele trei fetiţe (**1, 3, 4**) se vor afla în aceeaşi grupă. Fetiţele **2** şi **5** vor forma fiecare câte o grupă deoarece nu au cules flori de acelaşi tip cu nici una dintre celelalte fetiţe.

Soluţie problema 1

Se deschide fișierul NUMERE.IN. Se citeşte primul în variabila x număr apoi celelalte numere în variabila y. Se verifică dacă numerele consecutive sunt egale. Dacă sunt egale atunci se contorizează altfel se reinițializează contorul.

```
#include<fstream>
using namespace std;
ifstream f("NUMERE.IN");
int main()
{ long x,y,n;
  f>>x;
  n=1;
  while(f>>y)
    { if(x==y) n++;
      else { cout<<x<<" "<<n<<" ";
        n=1;
      }
```

```
        x=y;
      }
    cout<<x<<" "<<n;
    return 0;
}
```

Soluţie problema 2

Citim de pe prima linie a fişierului numărul de linii pe care sunt scrise note în variabila n, apoi notele le reţinem într-o matrice a cu n linii şi 3 coloane. Se calculează media pe fiecare linie şi se afişează în fişierul de ieşire ”MEDII.OUT”.

```
#include <fstream.h>
#include<iostream.h>
int n,m=3,i,j;
float a[20][20],s;
int main()
{
   ifstream f("NOTE.IN");
   f>>n;
   for(i=1;i<=n;i++)
         for(j=1;j<=3;j++)
               f>>a[i][j];
         ofstream g("MEDII.OUT");
         for(i=1;i<=n;i++)
         {
               s=0;
```

```
            for(j=1;j<=m;j++)
                  s+=a[i][j];
            g<<s/3<<endl;
      }
      f.close();
      g.close();
      return 1;
}
```

Soluţie problema 3

Deschidem cele două fişiere pentru citire, respectiv scriere, citim valoarea n de pe prima linie din fişierul ‘impare.in’, după care, citim pe rând, câte un număr din cele n aflate pe următoarea linie şi testăm dacă este impar. În caz afirmativ numărul este afişat în fişierul ‘impare.out’.

```
#include <fstream.h>
int n,i,x;
int main()
{
   ifstream f("impare.in");
   ofstream g("impare.out");
   f>>n;
   for(i=1;i<=n;i++)
   {
         f>>x;
         if(x%2!=0)
               g<<x<<" ";
   }
   return 1;
}
```

Soluţie problema 4

Vom copia elementele din fişier într-un vector a, a cărui dimensiune va fi k. În vectorul v vom reţine pe poziţie impară poziţia de început a unei secvenţe din vectorul a, şi pe poziţie pară, lungimea secvenţei. Parcurgem vectorul v şi calculăm maximul de pe poziţiile pare (maximul lungimii secvenţelor), după care vom afişa din vectorul a secvenţele de lungime maximă. Dacă pe poziţia i în vectorul v avem lungimea maximă a unei secvenţe, atunci a[i-1] reprezintă elementul de început al secvenţei de lungime maximă. Secvenţa de lungime maximă se află în intervalul (a[i-1], a[i-1]+max-1).

```
#include <iostream.h>
#include <fstream.h>
int x,max=0,i,a[50],v[20],k=0,nr,l=0,j;
int main()
{
   ifstream f("numere.txt");
   while(!f.eof())
   {
        f>>a[++k];
   }
   f.close();
   i=1;
```

```
    while(i<=k)
    {
          nr=0;
          v[++l]=i;
          while(a[i]!=0)
                {nr++;i++;}
          v[++l]=nr;
          i++;
    }
    for(i=2;i<=l;i+=2)
          if(v[i]>max)
                max=v[i];
    for(i=2;i<=l;i+=2)
                if(v[i]==max)
                {
                for(j=v[i-1];j<v[i-
1]+max;j++)
                      cout<<a[j]<<" ";
                    cout<<endl;
          }
          return 1;
```

```
}
```

Soluţie problema 5

Testăm mai întâi valoarea lui n. Dacă n este par atunci creştem valoarea lui n cu 1. Vom genera partea superioară a rombului, cele n/2+1 linii ce vor fi afişate în fişierul 'romb.out'. Vom afişa pe fiecare rând j spaţii (j=1, n/2*i-1), şi j caractere 'X ' (j=1, 2*i-1). În partea inferioară a rombului mai rămân de afişat n/2 linii în fişier, j caractere spaţiu (j=1, i), şi j caractere 'X ' (j=1, n-2*i).

```
#include <fstream.h>
#include <iostream.h>
int n,i,j;
int main()
{
   ifstream f("romb.in");
   ofstream g("romb.out");
   f>>n;
   if(n%2==0)
         n=n+1;
   for(i=1;i<=n/2+1;i++)
   {
         for(j=1;j<=n/2-i+1;j++)
               g<<" ";
```

```
            for(j=1;j<=2*i-1;j++)
                  g<<"X";
            g<<endl;
      }
      for(i=1;i<=n/2;i++)
      {
            for(j=1;j<=i;j++)
                  g<<" ";
            for(j=1;j<=n-2*i;j++)
                  g<<"X";
            g<<endl;
      }
      return 1;
}
```

Soluţie problema 6

Citim numerele din fişierul ’numere.in’ în vectorul v, care va avea după citirea elementelor n numere. Ordonăm crescător elementele din v, folosind metoda de sortare prin selecţie directă: comparăm primul element cu toate elementele de pe poziţiile 2, 3, …n şi aducem pe prima poziţie elementul cel mai mic din vector, comparăm apoi elementul de pe poziţia 2 cu elementele de pe poziţiile 3, 4, ….n şi elementul minim îl aşezăm pe poziţia 2, etc. Eliminăm aparţiile multiple ale elementelor din vectorul v, după care afişăm în fişierul ‘numere.out’, prin două parcurgeri ale vectorului v elementele pare pe prima linie şi elementele impare pe cea de- a doua linie.

```
#include <iostream.h>
#include <fstream.h>
int n=0, v[50],i,j,k,l;
int main()
{
   ifstream f("numere.in");
   ofstream g("numere.out");
   while(!f.eof())
   {
         f>>v[++n];
   }
```

```
        for(i=1;i<n;i++)
        for(j=i+1;j<=n;j++)
              if(v[i]>v[j])
              {
                    int aux=v[i];
                    v[i]=v[j];
                    v[j]=aux;
              }
              for(i=1;i<n;i++)
                    for(j=i+1;j<=n;j++)
                          if(v[i]==v[j])
                          {
for(l=i;l<n;l++)
v[l]=v[l+1];
                              n--;
                          }
              for(i=1;i<=n;i++)
                    if(v[i]%2==0)
                          g<<v[i]<<" ";
                    g<<endl;
```

```
                        for(i=1;i<=n;i++)
                              if(v[i]%2!=0)
                              g<<v[i]<<" ";
    return 1;
}
```

Soluţie problema 7

Deschidem fişierul ‘numere.in’ pentru citire şi celelalte două fişiere ’pare.out’ ’impare.out’ pentru scriere în fişier. Se citeşte câte un număr în variabila x, din primul fişier, se verifică dacă este par şi în caz afirmativ se scrie în fişierul ’pare.out’, altfel se scrie în fişierul ’impare.out’.

```
#include <iostream.h>
#include <fstream.h>
int n,i,x;
int main()
{
   ifstream f("numere.in");
   ofstream g("pare.out");
   ofstream h("impare.out");
   f>>n;
   for(i=1;i<=n;i++)
   {
         f>>x;
         if(x%2==0)
               g<<x<<" ";
         else
```

```
                h<<x<<" ";
    }
    f.close();
    g.close();
    h.close();
    return 1;
}
```

Soluţie problema 8

Citim de pe prima linie a fişierului 'matrice.in' dimensiunea n, linia k şi linia l, apoi elementele matricei. Interschimbăm elementele de pe coloana k cu elementele de pe coloana l şi afişăm matricea rezultată în fişierul 'matrice.out'.

```
#include <fstream.h>
#include<iostream.h>
int a[50][50],n,i,j,k,l;
int main()
{
   ifstream f("matrice.in");
   ofstream g("matrice.out");
   f>>n>>k>>l;
   for(i=1;i<=n;i++)
         for(j=1;j<=n;j++)
               f>>a[i][j];
         for(i=1;i<=n;i++)
         {
               int aux=a[i][k];
               a[i][k]=a[i][l];
```

```
            a[i][l]=aux;
        }
        for(i=1;i<=n;i++)
        {
            for(j=1;j<=n;j++)
                g<<a[i][j]<<" ";
            g<<endl;
        }
        return 1;
}
```

Soluţie problema 9

Se citesc cele n numere din fişierul de intrare “numere.in” în vectorul v. Pentru fiecare număr din vector se verifică dacă este strict mai mic decât 100. În caz afirmativ numărul are mai puţin de trei cifre şi poate fi afişat în fişierul “numere.out”.

```
#include <fstream.h>
int n,i,v[50];
int main()
{
   ifstream f("numere.in");
   f>>n;
   for(i=1;i<=n;i++)
         f>>v[i];
   ofstream g("numere.out");
   for(i=1;i<=n;i++)
         if(v[i]<100)
               g<<v[i]<<" ";
         f.close();
         g.close();
}
```

Soluţie problema 10

Se citesc numere din fişierul “numere.in” cât timp nu este sfârşit de fişier. Nu merele pozitive sunt reţinute în vectorul v, care se ordonează crescător, prin metoda sortării prin selecţie directă. Vectorul sortat crescător se afişează în fişierul “numere.out”.

```
#include <fstream.h>
int n=0,v[50],i,j,x;
int main()
{
   ifstream f("numere.in");
   while(!f.eof())
   {
         f>>x;
         if(x>0)
         v[++n]=x;

   }
   for(i=1;i<n;i++)
         for(j=i+1;j<=n;j++)
               if(v[i]>v[j])
```

```
                {
                        int aux=v[i];
                        v[i]=v[j];
                        v[j]=aux;
                }
                ofstream g("numere.out");
                for(i=1;i<=n;i++)
                        g<<v[i]<<" ";
                f.close();
                g.close();
    return 1;
}
```

Soluţie problema 11

Căutăm perechi de forma (i, j), cu i=1,n-1 şi j=i+1,n. Formăm aceste perechi în două cicluri, şi pentru fiecare pereche (i, j) determinăm cel mai mare divizor comun al elementelor perechii, după care testăm dacă rezultatul este egal cu valoarea d.

```
#include <fstream.h>
#include <iostream.h>
int n,d,i,j,x,y;
int main()
{
   ifstream f("divizori.in");
   ofstream g("divizori.out");
   f>>n>>d;
   for(i=1;i<n;i++)
         for(j=i+1;j<=n;j++)
         {
               x=i;y=j;
               while(x!=y)
                     if(x>y)
                           x=x-y;
```

```
                        else
                              y=y-x;
                        if(x==d)
                        g<<i<<" "<<j<<endl;
            }
            f.close();        g.close();
            return 1;
    }
```

Soluţie problema 12

Citim fiecare linie din fişierul de intrare ‘fractii.in’ în variabilele x, respectiv z, corespunzătoare numitorului şi numărătorului. Calculăm cel mai mare divizor comun dintre cele două numere, şi scriem fracţia simplificată în fişierul ’fracţii.out’ prin împărţirea celor două numere la cel mai mare divizor comun.

```
#include <iostream.h>
#include <fstream.h>
int x,y,d,p,q;
int main()
{
   ifstream f("fractii.in");
   ofstream g("fractii.out");
   while(!f.eof())
   {
         f>>x>>y;
         p=x;q=y;
         while(p!=q)
               if(p>q)
                     p=p-q;
               else
```

```
                    q=q-p;
                g<<x/p<<" "<<y/p<<endl;
    }
    f.close();  g.close();
    return 1;
}
```

Soluţie problema 13

Citim din fişierul ‘litera.in’ numărul de caractere în variabila n, apoi citim pe fiecare din cele n linii caracterul în variabila a şi numărul corespunzător în variabila x. Afişăm în fişierul ’litera.out’ pe fiecare linie caracterul citit de x ori.

```
#include <iostream.h>
#include <fstream.h>
int n,i,x,j;
char a;
int main()
{
   ifstream f("litera.in");
   ofstream g("litera.out");
   f>>n;
   for(i=1;i<=n;i++)
   {
         f>>a>>x;
         for(j=1;j<=x;j++)
               g<<a;
         g<<endl;
```

```
        }
        f.close();
        g.close();
        return 1;
    }
```

Soluţie problema 14

Vom parcurge liniile ce se vor afişa, cu o variabilă i (i=1,n), şi folosim o variabilă k pe care o iniţializăm la fiecare început de linie cu 1. Pe fiecare linie i scriem i numre impare.

```
#include<iostream.h>
#include<fstream.h>
int n,i,j,k=0;
int main()
{
   ofstream f("triunghi.txt");
   cout<<"n=";cin>>n;
   for(i=1;i<=n;i++)
   {          k=1;
         for(j=1;j<=i;j++)
         {
               f<<k<<" ";
               k=k+2;
         }
         f<<endl;
   }
```

```
        f.close();
        return 1;
    }
```

Soluţie problema 15

Din primul fişier “sir.in” se citeşte primul număr şi se afişează în fişierul ”sir.out”. Cât timp nu este sfârşit de fişier se citeşte câte un număr din fişierul de intrare în variabila z se verifică dacă este diferit de numărul citit anterior în variabila x, şi în caz afirmativ se afişează în fişierul ”sir.out”. Datorită faptului că numerele sunt scrise în ordine crescătoare în fişier în momentul în care s-au găsit două numere diferite în parcurgerea şirului se trece la un nou număr ce trebuie afişat.

```
#include <fstream.h>
int n,i,x,y;
int main()
{
   ifstream f("sir.in");
   ofstream g("sir.out");
   f>>x;
   g<<x<<" ";
   while(!f.eof())
   {
         f>>y;
         if(y!=x)
               g<<y<<" ";
```

```
            x=y;
        }
        f.close();
        g.close();
        return 1;
    }
```

Soluţie problema 16

Cât timp nu este sfârşit de fişier se citeşte câte un număr în variabila x, se extrag cifrele de la ultima la prima cifră şi dacă este impară se adaugă la un nou număr n. Variabila n va reţine numărul format doar din cifrele impare ale numărului x. n se resetează pentru fiecare nou număr x citit. După ce s-a construit n se verifică dacă este diferit de 0, şi în caz afirmativ se afişează în fişierul "numere.out".

```
#include <fstream.h>
int n,x,i,p;
int main()
{
   ifstream f("numere.in");
   ofstream g("numere.out");
   while(!f.eof())
   {
         p=1;
         f>>x;
         n=0;
         while(x)
         {
```

```
                if(x%2!=0)
                {
                        n=(x%10)*p+n;
                        p=p*10;
                }
                x=x/10;
        }
        if(n)
                g<<n<<" ";
    }

    return 1;
}
```

Soluţie problema 17

Citim din fişierul de intrare ’matrice.in’ dimensiunea n şi elementele matricii. Pentru parcurgerea elementelor de sub diagonala principală, inclusiv elementele situate pe diagonală vom proceda astfel: parcurgem liniile cu un indice i (i=1,n), şi pentru elementele de pe coloane folosim un indice j (j=1,i).

```
#include <iostream.h>
#include <fstream.h>
int a[50][50],n,i,j;
int main()
{
   ifstream f("matrice.in");
   ofstream g("matrice.out");
   f>>n;
   for(i=1;i<=n;i++)
         for(j=1;j<=n;j++)
               f>>a[i][j];
         for(i=1;i<=n;i++)
         {
               for(j=1;j<=i;j++)
                     g<<a[i][j]<<" ";
```

```
                g<<endl;
            }
            return 1;
    }
```

Soluţie problema 18

Se foloseşte un vector al frecvenţei apariţiilor cifrelor în toate numerele din fişier. Spre exemplu pe poziţia i=4, avem valoare 5, acest lucru reprezentând faptul că cifra 4 apare de 5 ori în toate numerele din fişier. Vectorul se iniţioalizează cu 0 la început. În final vor fi afişate cifrele din vector de la 9 la 0 cele care apar şi de câte ori apar.

```
#include <fstream.h>
int n,v[10],i,x,j;
int main()
{
   ifstream f("numere.in");
   for(i=0;i<=9;i++)
         v[i]=0;
   while(!f.eof())
   {
         f>>x;
         while(x)
         {
               v[x%10]++;
               x=x/10;
```

```
            }
      }
      ofstream g("numere.out");
      for(i=9;i>=0;i--)
            for(j=1;j<=v[i];j++)
                  g<<i;
            return 1;
}
```

Soluţie problema 19

Vom folosi un vector al apariţiei cifrelor într-un număr v. Pentru fiecare număr citit din fişierul de intrare se reţine în vector frecvenţa apariţiei cifrelor. Dacă o cifră apare de cel puţin două ori atunci numărul nu va fi afişat în fişierul de ieşire.

```
#include <fstream.h>
int n,i,v[10],x,p,y,j;
int main()
{
   ifstream f("numar.in");
   ofstream g("numar.out");
   f>>n;
   for(i=0;i<=9;i++)
         v[i]=0;
   for(i=1;i<=n;i++)
   {
         f>>x;
         y=x;
         while(x)
         {
```

```
                v[x%10]++;
                x=x/10;
            }
            p=1;
            for( j=0;j<=9;j++)
                if(v[j]!=0)
                    if(v[j]>1)
                        p=0;
                    if(p)
                    g<<y<<" ";
                    for(j=0;j<=9;j++)
                        v[j]=0;
    }
            return 1;
}
```

Soluţie problema 20

Citim elementele din fişier într – o matrice a, având următoarea semnificaţie: pe linia i se găsesc tipurile de flori din buchetul fetiţei i. Vom utiliza doi vectori viz şi v ale căror elemente sunt 0 şi 1, 0 dacă fetiţa de pe poziţia i nu a fost adăugată la un grup, 1 dacă fetiţa de pe poziţia i a fost adăugată la un grup. Vectorul viz, va reţine toate fetiţele adăugate la grupuri până la un moment dat, pentru a se evita alegerea unei fetiţe ce a mai fost selecată, vectorul v va fi resetat pentru fiecare grup format. Se parcurge fiecare linie i din matrice, dacă nu a fost parcursă şi se compară element cu element cu toate celelalte elemente de pe linia j (j=i+1,n) şi dacă se găseşte cel puţin un element comun linia se adaugă la grup, evident se marchează în vectorii v şi viz cu 1 pe poziţia liniei. Fiecare grup format, va fi afişat în fişierul de ieşire 'flori.out'.

```
#include <iostream.h>
#include <fstream.h>
int a[50][50],n,k,i,j,viz[50],v[50],p,q;
int main()
{
   ifstream f("flori.in");
   ofstream g("flori.out");
   f>>n>>k;
   for(i=1;i<=n;i++)
```

```
        {
                viz[i]=0;
                v[i]=0;
        }
for(i=1;i<=n;i++)
        for(j=1;j<=k;j++)
                f>>a[i][j];
        for(i=1;i<=n;i++)
        {
                for(j=1;j<=n;j++)
                        v[j]=0;
                if(viz[i]==0)
                {       viz[i]=1;
                v[i]=1;
                for(j=1;j<=k;j++)
                        for(p=i+1;p<=n;p++)
                                if(v[p]==0)
                                {

for(q=1;q<=k;q++)

if(a[i][j]==a[p][q])
```

```
                                        {
v[p]=1;
viz[p]=1;
                                        }
                                  }
                  }
                  int p=0;
                  for(j=1;j<=n;j++)
                        if(v[j]==1)
                        {     p=1;
                              g<<j<<" ";
                        }
                        if(p)
                              g<<endl;
            }
            return 1;
}
```

BIBLIOGRAFIE

- *George Daniel Mateescu, Pavel Florin Moraru – Informatica, fișe de lucru pentru elevi, clasele a IX-a și a X-a, Editura Donaris, Sibiu, 2006.*
- *Carmen Popescu – Culegere de probleme de informatică, Editura Donaris, Sibiu 2002.*
- *Emanuela Cerchez – Informatică – Culegere de probleme pentru liceu, Editura Polirom, Iasi, 2005.*
- *Doina Hrinciuc Logofătu – C++ - probleme rezolvate și algoritmi, Editura Polirom, Iași, 2001.*
- *Pavel Florin Moraru – Bacalaureat Informatică, Editura Petrion, București, 2000.*
- *B. Pătruț – Aplicații C și C++, Editura Teora, București, 1998.*
- *D Stoilescu – Culegere de C/C++, Editura Radial, Galați, 1998.*
- *Mariana Miloșescu – Informatică intensiv – Manual pentru clasa a XI-a, Editura Didactică și Pedagogică, R. A., București*