**4Pb\_Atestat\_C++\_Rezolvate**

**Bilet 40**

Se citesc două triplete de numere naturale (**d1**,**m1**,**y1**), respectiv (**d2**,**m2**,**y2**). Să se verifice dacă cele două triplete reprezintă două date calendaristice valide, în care **d1,d2** = ziua (day), **m1,m2** = luna (month), **y1,y2** = anul (year). Se consideră an bisect orice an care fie este divizibil cu 400, fie este divizibil cu 4 dar nu şi cu 100.

Dacă cele două triplete reprezintă două date calendaristice valide, atunci în ipoteza că prima dată reprezintă ziua de naştere a unei persoane, iar a doua dintre ele reprezintă data curentă, să se determine, în ani împliniţi, vârsta persoanei la data curentă.

Se garantează că prima dintre date este anterioară celei de-a doua.

Exemplu

| 29 2 2001  30 5 2013 | data invalida |
| --- | --- |
| 25 4 1995  20 5 2013 | 18 ani |
| 25 6 1995  10 5 2013 | 17 ani |

**VARIANTA 1**

**#include <iostream>**

**#include <fstream>**

**using namespace std;**

**struct Data**

**{**

**int zi;**

**int luna;**

**int an;**

**};**

**bool checkDate(const Data& data)**

**{**

**bool invalidDate = false;**

**switch (data.luna)**

**{**

**case (1): if (data.zi > 31) invalidDate = true; break;**

**case (2):**

**if (data.an % 400 == 0 || (data.an % 4 == 0 && data.an % 100 != 0))**

**{**

**if (data.zi > 29) invalidDate = true;**

**}**

**else if (data.zi > 28) invalidDate = true;**

**case (3): if (data.zi > 31) invalidDate = true; break;**

**case (4): if (data.zi > 30) invalidDate = true; break;**

**case (5): if (data.zi > 31) invalidDate = true; break;**

**case (6): if (data.zi > 30) invalidDate = true; break;**

**case (7): if (data.zi > 31) invalidDate = true; break;**

**case (8): if (data.zi > 31) invalidDate = true; break;**

**case (9): if (data.zi > 30) invalidDate = true; break;**

**case (10): if (data.zi > 31) invalidDate = true; break;**

**case (11): if (data.zi > 30) invalidDate = true; break;**

**case (12): if (data.zi > 31) invalidDate = true; break;**

**default: invalidDate = true;**

**}**

**return !invalidDate;**

**}**

**int main()**

**{**

**std::ifstream f("atestat.txt");**

**Data startDate, endDate;**

**f >> startDate.zi >> startDate.luna >> startDate.an;**

**f >> endDate.zi >> endDate.luna >> endDate.an;**

**if (!checkDate(startDate) || !checkDate(endDate))**

**{**

**cout << "Data invalida";**

**return 0;**

**}**

**else**

**{**

**int ani = endDate.an - startDate.an;**

**if(endDate.luna == startDate.luna)**

**{**

**if (endDate.zi <= startDate.zi)**

**ani--;**

**}**

**else if(endDate.luna < startDate.luna)**

**ani--;**

**cout << ani;**

**}**

**}**

**VARIANTA 2**

**#include<fstream>**

**#include<iostream>**

**#include <string.h>**

**using namespace std;**

**int verif(int d,int m,int y)**

**{**

**if(m==2&&d>28) return 0;**

**if(m==4 && d>30) return 0;**

**if(m==6&&d>30)return 0;**

**if(m==9&&d>30)return 0;**

**if(m==11&&d>30)return 0;**

**if(d<1 || d>31) return 0;**

**if(m<1 || m>12) return 0;**

**return 1;**

**}**

**int main()**

**{**

**int d1,m1,y1,d2,m2,y2;**

**cin>>d1>>m1>>y1>>d2>>m2>>y2;**

**if(verif(d1,m1,y1)&&verif(d2,m2,y2))**

**{**

**int ani=y2-y1;**

**if(m1>m2) ani--;**

**else**

**if(m1==m2 && d1>d2) ani--;**

**cout<<ani;**

**}**

**else**

**cout<<"date invalide";**

**return 0;**

**}**

**BILET 39**

Fișierele X.txt si Y.txt conțin fiecare numele a 7 persoane, câte un nume pe fiecare linie având cel mult 25 caractere. Știind că, în fiecare fișier numele sunt memorate în ordine alfabetică, scrieți un program care să citească din cele două fișiere numele și să afișeze pe ecran toate numele din cele două fișiere în ordine alfabetică, separate printr-un singur spațiu. Dacă un nume se află în ambele fișiere, atunci el se va afișa o singură dată.

Exemplu

| **X.txt**  Ana  Dana  Daniel  Ene  Mara  Nae  Paul | **Y.txt**  Angi  Cora  Dora  Horia  Oana  Paul  Tibi | Ana Angi Cora Dana Daniel Dora Ene Horia Mara Nae Oana Paul Tibi |
| --- | --- | --- |

**#include <iostream>**

**#include <fstream>**

**#include <cstring>**

**using namespace std;**

**int main()**

**{**

**ifstream xin("X.txt");**

**ifstream yin("Y.txt");**

**char x[26], y[26];**

**int i = 0, j = 0;**

**xin >> x;**

**yin >> y;**

**while(i < 7 && j < 7)**

**{**

**if (strcmp(x, y) < 0)**

**{**

**cout << x << " ";**

**i++;**

**xin >> x;**

**}**

**else if(strcmp(x, y) > 0)**

**{**

**cout << y << " ";**

**j++;**

**yin >> y;**

**}**

**else if (strcmp(x, y) == 0)**

**{**

**cout << x << " ";**

**i++;**

**j++;**

**xin >> x;**

**yin >> y;**

**}**

**}**

**if (i < 7)**

**{**

**while(i < 7)**

**{**

**xin >> x;**

**cout << x << " ";**

**i++;**

**}**

**}**

**if (j < 7)**

**{**

**while(j < 7)**

**{**

**yin >> y;**

**cout << y << " ";**

**j++;**

**}**

**}**

**}**

**Biletul nr. 38**

Se citeşte din fișierul “text.in” un şir cu maxim 255 de caractere, format doar din litere (mici şi mari) şi spaţii. Se consideră cuvânt orice secvenţă formată doar din litere adiacente. Cuvintele sunt separate prin exact un spaţiu. Să se afişeze cuvintele şirului dat, în ordinea crescătoare a lungimilor lor, câte un cuvânt pe fiecare linie a ecranului.

Exemplu

| **text.in**  examen de atestat profesional la informatica | de  la  examen  atestat  profesional  informatica |
| --- | --- |

**#include <iostream>**

**#include <cstring>**

**#include <fstream>**

**using namespace std;**

**ifstream f("text.in");**

**int main()**

**{**

**char s[256],a[256][256],\*p,aux[256];**

**int n=0;**

**f.getline(s,256);**

**p=strtok(s," ");**

**while(p)**

**{**

**strcpy(a[n++],p);**

**p=strtok(NULL," ");**

**}**

**for(int i=0; i<n-1; i++)**

**{**

**for(int j=i+1; j<n; j++)**

**if(strlen(a[i])>strlen(a[j]))**

**{**

**strcpy(aux,a[i]);**

**strcpy(a[i],a[j]);**

**strcpy(a[j],aux);**

**}**

**}**

**for(int i=0; i<n; i++)**

**cout<<a[i]<<endl;**

**return 0;**

**}**

**Bilet 37**

Se citeşte din fișierul “text.in” un şir cu maxim 255 de caractere, format doar din litere (mici şi mari) şi spaţii. Se consideră cuvânt orice secvenţă formată doar din litere adiacente. Cuvintele sunt separate prin exact un spaţiu. Să se afişeze cuvintele şirului dat, în ordine alfabetică.

Exemplu

| **text.in**  anul trecut aveam alte prioritati | alte anul aveam prioritati trecut |
| --- | --- |

**#include <iostream>**

**#include <cstring>**

**#include <fstream>**

**using namespace std;**

**ifstream f("text.in");**

**int main()**

**{**

**char s[256],a[256][256],\*p,aux[256];**

**int n=0;**

**cin.getline(s,256);**

**p=strtok(s," ");**

**while(p)**

**{**

**strcpy(a[n++],p);**

**p=strtok(NULL," ");**

**}**

**for(int i=0; i<n-1; i++)**

**{**

**for(int j=i+1; j<n; j++)**

**if(strcmp(a[i],a[j])>0)**

**{**

**strcpy(aux,a[i]);**

**strcpy(a[i],a[j]);**

**strcpy(a[j],aux);**

**}**

**}**

**for(int i=0; i<n; i++)**

**cout<<a[i]<<endl;**

**return 0;**

**}**

**Biletul nr. 36**

Se citeşte din fișierul “text.in” un șir de maxim 255 de caractere. Să se determine cea mai lungă secvență de cifre alăturate din șir. Secvența găsită se va afișa pe ecran.

Exemplu

| **text.in**  A12abac12223defg895 | 12223 |
| --- | --- |

**#include <iostream>**

**#include <cstring>**

**#include <fstream>**

**using namespace std;**

**ifstream f("text.in");**

**int main()**

**{**

**char c[256];**

**f.getline(c,256);**

**int nr=0,nrmax=-1,poz;**

**int n=strlen(c);**

**for(int i=0; i<n; i++)**

**{**

**if(c[i]>='0' && c[i]<='9')**

**nr++;**

**else if(nr>nrmax)**

**{**

**nrmax=nr;**

**poz=i-1;**

**nr=0;**

**}**

**}**

**if(nr>nrmax)**

**{nrmax=nr;**

**poz=n-1;}**

**for(int i=poz-nrmax+1;i<=poz;i++)**

**cout<<c[i];**

**}**

**Biletul nr. 35**

Se citeşte din fișierul “text.in” un şir cu maxim 255 de caractere, format doar din litere mici şi spaţii. Se consideră cuvânt orice secvenţă formată doar din litere adiacente. Cuvintele sunt separate prin exact un spaţiu. Să se afișeze în fișierul “rime.out” toate perechile de cuvinte care rimează. Două cuvinte rimează dacă au ultimele două litere identice.

Exemplu

| **text.in**  zic cei de la sate ca soarele puternic ne arde | **rime.out**  zic puternic  de arde |
| --- | --- |

**#include <iostream>**

**#include <fstream>**

**#include <cstring>**

**using namespace std;**

**ifstream f("text.in");**

**ofstream g("rime.out");**

**int main()**

**{**

**char text[256],\*p, a[256][256];**

**int i=0;**

**f.getline(text,256);**

**p=strtok(text," ");**

**while (p)**

**{**

**strcpy(a[i++],p);**

**p=strtok(NULL," ");**

**}**

**for(int j=0; j<i-1; j++)**

**{**

**int n=strlen(a[j]);**

**for(int k=j+1; k<i; k++)**

**{**

**int m=strlen(a[k]);**

**if(a[j][n-1]==a[k][m-1] && a[j][n-2]==a[k][m-2])**

**g<<a[j]<<" "<<a[k]<<" ";**

**}**

**}**

**return 0;**

**}**

**Biletul nr. 34**

Se citeşte din fișierul “text.in” un şir cu maxim 255 de caractere, format doar din litere mici şi spaţii. Se consideră cuvânt orice secvenţă formată doar din litere adiacente. Cuvintele sunt separate prin exact un spaţiu. Să se afișeze cuvintele palindrom din text. Dacă nu există cuvinte palindrom atunci se va afișa mesajul “NU EXISTĂ”.

Exemplu

| **text.in**  aerisirea este blocata de un cojoc | aerisirea  cojoc |
| --- | --- |

**#include <iostream>**

**#include <cstring>**

**#include <fstream>**

**using namespace std;**

**//ifstream f("text.in");**

**int main()**

**{**

**char a[256] ,b[256],c[256], \*p;**

**int i,n;**

**cin.getline(a,256);**

**p=strtok(a," ");**

**while(p)**

**{**

**strcpy(b,p);**

**strcpy(c,p);**

**strrev(b);**

**if(strcmp(b,c)==0) cout<<b<<" ";**

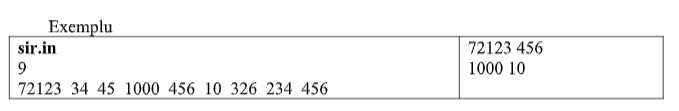
**p=strtok(NULL, " ");**

**}**

**}**

**Biletul nr. 33**

**Se citeşte din fişierul “sir.in”, de pe prima linie, un număr natural n (1≤n≤50). De pe următoarea linie se citesc n numere naturale de maxim 9 cifre. Să se afişeze toate perechile de elemente egal depărtate de mijloc, care au aceeași suma a cifrelor.**

****

**#include <iostream>**

**#include <fstream>**

**using namespace std;**

**ifstream f("sir.in");**

**int suma(int n)**

**{**

**int s=0;**

**while (n)**

**{**

**s=s+n%10;**

**n/=10;**

**}**

**return s;**

**}**

**int main()**

**{int n,v[100];**

**f>>n;**

**for(int i=0;i<n;i++)**

**f>>v[i];**

**for(int i=0;i<n/2;i++)**

**if(suma(v[i])==suma(v[n-i-1]))**

**cout<<v[i]<<" "<<v[n-i-1]<<endl; }**

**Biletul nr. 32**

**Se citesc din fişierul “numere.in” doua numere naturale de maxim 9 cifre. Se cere să se afișeze pe ecran mesajul “DA“, dacă cele două numere sunt termeni consecutivi ai șirului lui Fibonacci,respectiv mesajul “NU“, daca cele două numere nu sunt termeni consecutivi ai șirului luiFibonacci. (Șirul lui Fibonacci 1,1,2,3,5,8,13,…).**

**Exemplu**

**numere.in**

**8 5 DA**

**21 13 DA**

**21 10 NU**

**#include <iostream>**

**#include <fstream>**

**using namespace std;**

**int v[10000];**

**void fibo()**

**{**

**v[0]=1,v[1]=1;**

**for(int i=2;i<10000;i++)**

**{**

**v[i]=v[i-1]+v[i-2];**

**}**

**}**

**int main()**

**{int x,y,ok;**

**ifstream f("numere.in");**

**f>>x>>y;**

**fibo();**

**for(int i=0;i<10000;i++)**

**{**

**if((x==v[i]&&y==v[i+1])||(x==v[i+1]&&y==v[i]))**

**{ok=1;break;}**

**else ok=0;**

**}**

**if (ok==1) cout<<"da";**

**else cout<<"nu";**

**return 0;**

**}**

**SAU**

**#include <iostream>**

**#include <fstream>**

**using namespace std;**

**int main()**

**{**

**ifstream fin("bac.txt");**

**int ls, ld;**

**while(fin >> ls >> ld)**

**{**

**if (ld < ls)**

**{**

**int aux = ld;**

**ld = ls;**

**ls = aux;**

**}**

**///8 5**

**int ok = 1;**

**while(ok)**

**{**

**cout << ls << " " << ld << endl;**

**int aux = ls;**

**ls = ld - ls;**

**ld = aux;**

**if (ls == 1 && ld == 1)**

**{**

**cout << "DA" << endl;**

**ok = 0;**

**}**

**else if (ls < 1 || ld < 1)**

**{**

**cout << "NU" << endl;**

**ok = 0;**

**}**

**}**

**}**

**}**

**Biletul nr. 31**

**Se citeşte din fişierul “sir.in”, de pe prima linie, un număr natural n (1≤n≤50). De pe următoarealinie se citesc n numere naturale de maxim 9 cifre. Să se scrie un program care să afișeze pe ecran câte din elementele șirului sunt termeni ai șirului lui Fibonacci. (Șirul lui Fibonacci 1,1,2,3,5,8,13,…).**

Exemplu

| **sir.in**  7  10 2 17 8 1 9 1 | 4 |
| --- | --- |

**#include <iostream>**

**#include <fstream>**

**using namespace std;**

**ifstream fin("sir.in");**

**int v[10000];**

**void fibo()**

**{**

**v[0]=1,v[1]=1;**

**for(int i=2;i<10000;i++)**

**{**

**v[i]=v[i-1]+v[i-2];**

**}**

**}**

**int main()**

**{**

**int n, nr=0;;**

**fin>>n;**

**fibo();**

**for(int i=0;i<n;i++)**

**{ int x;**

**fin>>x;**

**for(int j=0;j<10000;j++)**

**{**

**if(x==v[j])**

**{nr++;**

**break;}**

**}**

**}**

**cout<<nr;**

**return 0;**

**}**

**Biletul nr. 30**

Se citesc de la tastatură două cuvinte de maxim 20 de litere. Să se scrie un program care verifică dacă unul este anagrama celuilalt. Un cuvânt este anagramă pentru altul dacă cele două cuvinte conţin aceleaşi litere de acelaşi număr de ori, dar nu neapărat în aceeaşi ordine.

Exemplu

| mare  rame | Da |
| --- | --- |
| mare  mere | Nu |

**#include<iostream.h>**

**#include<string.h>**

**using namespace std;**

**int nr1[256],nr2[256];**

**int main()**

**{**

**char s1[21],s2[21];**

**cin>>s1>>s2;**

**int i;**

**for(i=0;i<strlen(s1);i++)**

**nr1[s1[i]]++;**

**for(i=0;i<strlen(s2);i++)**

**nr2[s2[i]]++;**

**int anagr=1;**

**for(i=0;i<=255;i++)**

**if(nr1[i]!=nr2[i]) anagr=0;**

**if(anagr) cout<<"Sunt anagrame";**

**else cout<<"NU sunt anagrame";**

**return 0;**

**}}**

**Biletul nr. 29**

Se citesc de la tastatură numitorii şi numărătorii (numere întregi de maxim 9 cifre) a două fracţii. Să se scrie un program care simplifică suma celor două fracţii şi afişează pe ecran numitorul şi numărătorul fracţiei rezultate.

Exemplu

| 1 2 1 3 | 5 6 |
| --- | --- |

**int cmmdc (int a, int b)**

**{**

**int r=0;**

**while (b!=0)**

**{**

**r=a%b;**

**a=b;**

**b=r;**

**}**

**return a;**

**}**

**int main()**

**{**

**int x1,y1,x2,y2, xs,ys;**

**cin>>x1>>y1;**

**cin>>x2>>y2;**

**xs=x1\*y2+x2\*y1;**

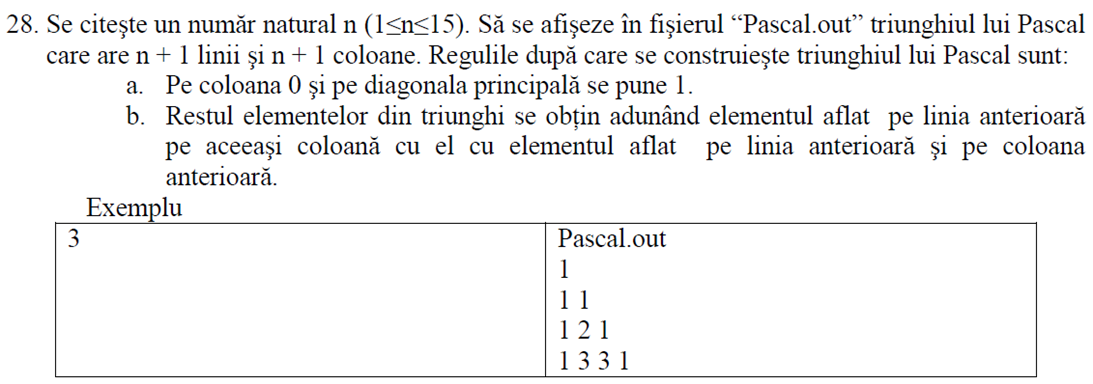
**ys=y1\*y2;**

**int aux=cmmdc(xs,ys);**

**cout<<xs/aux<<"/"<<ys/aux;**

**}**

**Bilet 28**

****

**int main()**

**{**

**ofstream f("pascal.out");**

**int n, a[16][16]={0};**

**cin>>n;**

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

**for(int j=1;j<=i;j++)**

**{**

**if(j==1 || i==j) a[i][j]=1;**

**else a[i][j]=a[i-1][j]+a[i-1][j-1];**

**}**

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

**{**

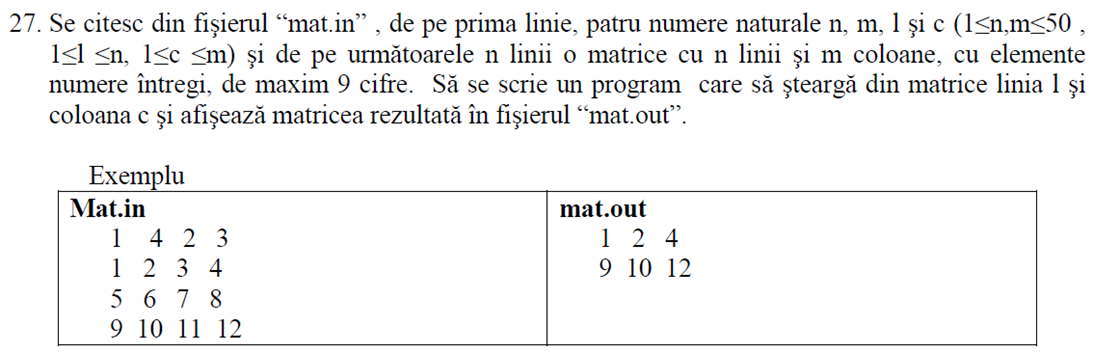
**for(int j=1;j<=n+1;j++)**

**f<<a[i][j]<<" ";**

**f<<endl;**

**}**

**}**

****

**#include<fstream.h>**

**#include<iostream.h>**

**using namespace std;**

**int main()**

**{**

**ifstream fin("mat1.in");**

**ofstream fout("mat1.out");**

**int a[51][51],n,m,i,j,l,c;**

**fin>>n>>m>>l>>c;**

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

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

**fin>>a[i][j];**

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

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

**a[i-1][j]=a[i][j];**

**n--;**

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

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

**a[i][j-1]=a[i][j];**

**m--;**

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

**{**

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

**fout<<a[i][j]<<" ";**

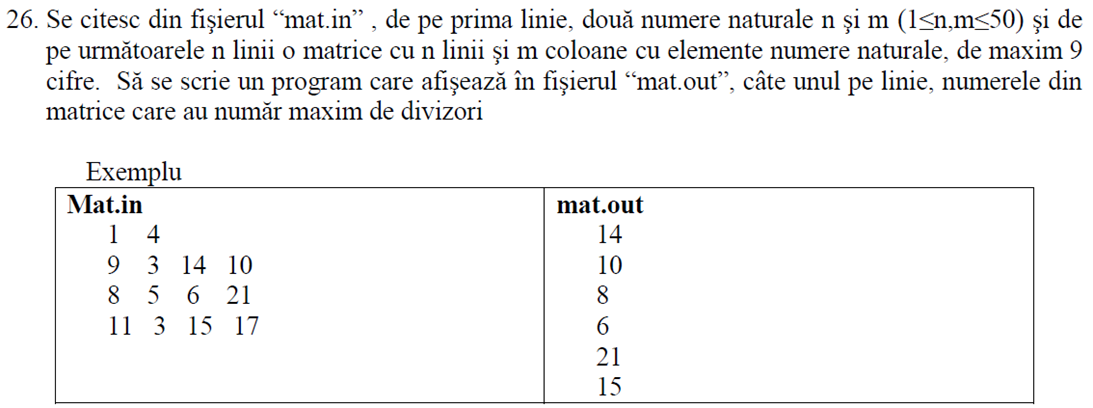
**fout<<"\n";**

**}**

**fin.close();fout.close();**

**return 0;**

**}**

****

**#include<fstream>**

**#include<iostream>**

**using namespace std;**

**int nrdiv(int x)**

**{**

**int nrdivizor=0, d;**

**for( d=1; d\*d<x; d++)**

**if(x%d==0)**

**nrdivizor+=2;**

**if(d\*d==x)**

**nrdivizor++;**

**return nrdivizor;**

**}**

**int main()**

**{**

**ifstream fin("mat.in");**

**ofstream fout("mat.out");**

**int a[50][50],n,m;**

**fin>>n>>m;**

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

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

**fin>>a[i][j];**

**int maxdiv=-1;**

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

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

**{**

**int nr=nrdiv(a[i][j]);**

**if(nr>maxdiv)**

**maxdiv=nr;**

**}**

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

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

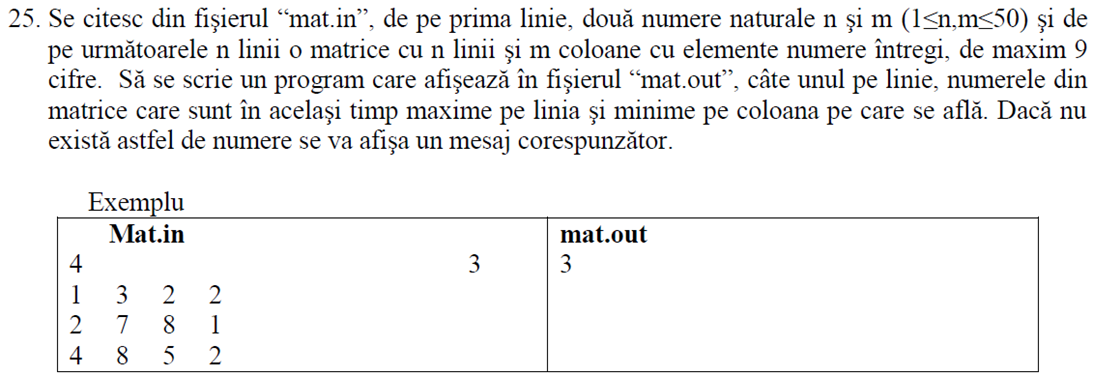
**{**

**if(nrdiv(a[i][j])==maxdiv)**

**fout<<a[i][j]<<endl;**

**}**

**}**

****

**#include <fstream>**

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**int v[51][51], n, m, minc, maxl, ok,i,j;**

**ifstream f("mat.in");**

**ofstream g("mat.out");**

**f>>n;**

**f>>m;**

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

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

**f>>v[i][j];**

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

**{**

**maxl=v[i][1];**

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

**if (v[i][j]>maxl)**

**maxl=v[i][j];**

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

**if(v[i][j]==maxl)**

**{**

**ok=1;**

**for(int k=1; k<=n; k++)**

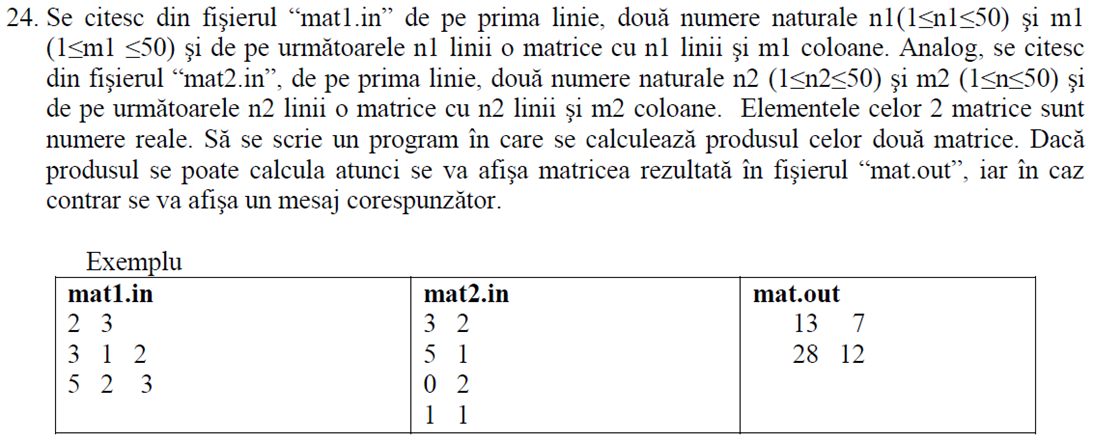
**if(v[k][j]<v[i][j])**

**ok=0;**

**if (ok==1)**

**cout<<v[i][j]<<" "; } }**

**return 0; }**

****

**#include<fstream.h>**

**#include<iostream.h>**

**using namespace std;**

**int main()**

**{**

**float a[51][51], b[51][51], c[51][51];**

**int n, m, m1,p,i,j,ok=1;**

**ifstream f1("mat1.in"), f2("mat2.in");**

**ofstream fout("mat.out");**

**f1>>n>>m;**

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

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

**f1>>a[i][j];**

**f1.close();**

**f2>>m1>>p;**

**for(i=1;i<=m1;i++)**

**for(j=1;j<=p;j++)**

**f2>>b[i][j];**

**f2.close();**

**if(m!=m1)**

**fout<<"Produsul nu se poate calcula!";**

**else**

**{**

**int k;**

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

**for(j=1;j<=p;j++)**

**{**

**c[i][j]=0;**

**for(k=1;k<=m;k++)**

**c[i][j]+=a[i][k]\*b[k][j];**

**}**

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

**{**

**for(j=1;j<=p;j++)**

**fout<<c[i][j]<<" ";**

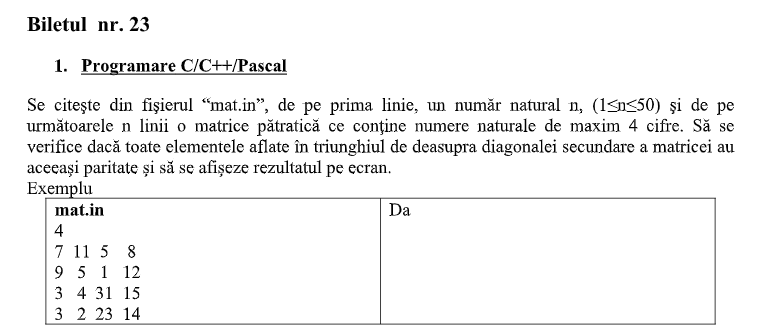
**fout<<"\n";**

**}**

**}**

**fout.close();**

**}**

****

**#include <iostream>**

**#include <fstream>**

**using namespace std;**

**int main()**

**{**

**int a[51][51], n, ok=1;**

**ifstream f("mat.in");**

**ofstream g("mat.out");**

**f>>n;**

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

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

**f>>a[i][j];**

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

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

**if(a[1][1]%2!=a[i][j]%2)**

**{**

**ok=0;**

**break;**

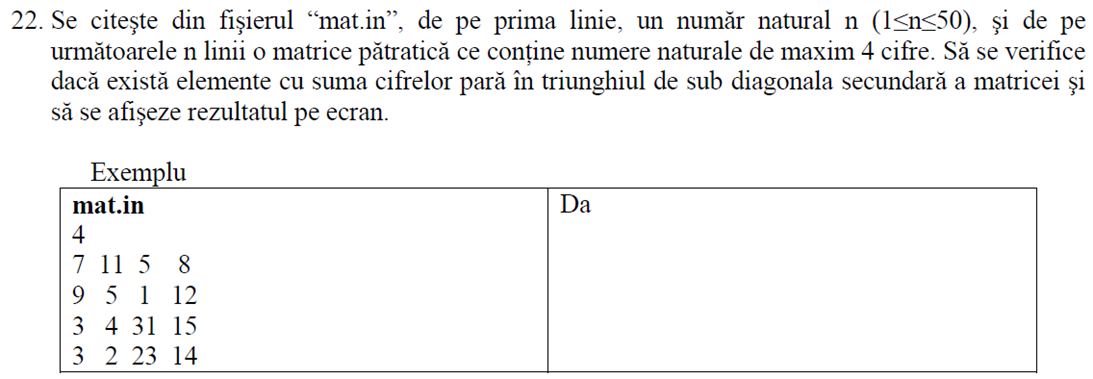
**}**

**if (ok==1) g<<"Da";**

**else g<<"Nu";**

**return 0;**

**}**

****

**#include <iostream>**

**#include <fstream>**

**using namespace std;**

**ifstream f("mat.in");**

**int sum(int x)**

**{**

**if(x==0) return 0;**

**return x%10+sum(x/10);**

**}**

**int main()**

**{**

**int n,ok=0;**

**f>>n;**

**int a[51][51];**

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

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

**{**

**f>>a[i][j];**

**}**

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

**{**

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

**{**

**if(j+i>n+1) if(sum(a[i][j])%2==0) ok=1;**

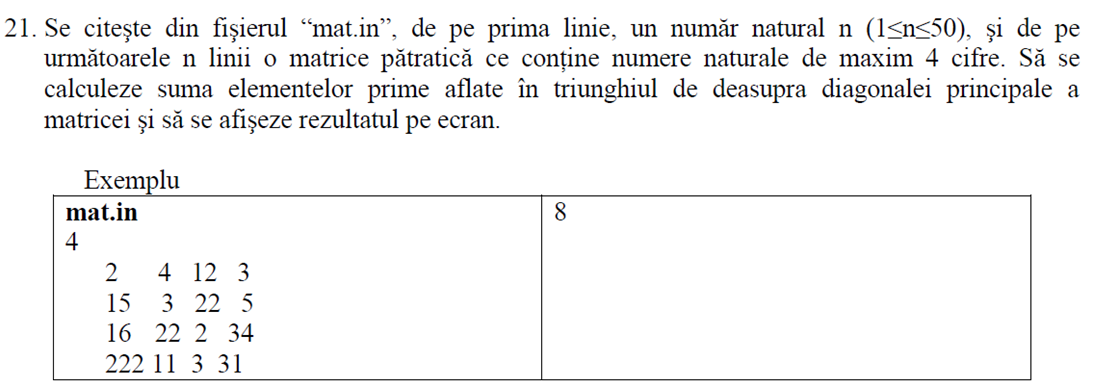
**}**

**}**

**if(ok==1) cout<<"DA";**

**else cout<<"NU";**

**}**

****

**#include <iostream>**

**#include <fstream>**

**using namespace std;**

**ifstream fin("mat.in");**

**int prim(int x)**

**{**

**if(x<2) return 0;**

**if(x!=2 && x%2==0) return 0;**

**for(int d=3;d\*d<=x;d+=2)**

**if(x%d==0)**

**return 0;**

**return 1;**

**}**

**int main()**

**{**

**int a[51][51],n,s=0;**

**fin>>n;**

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

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

**fin>>a[i][j];**

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

**for(int j=i+1;j<=n;j++)**

**{**

**if(prim(a[i][j])==1)**

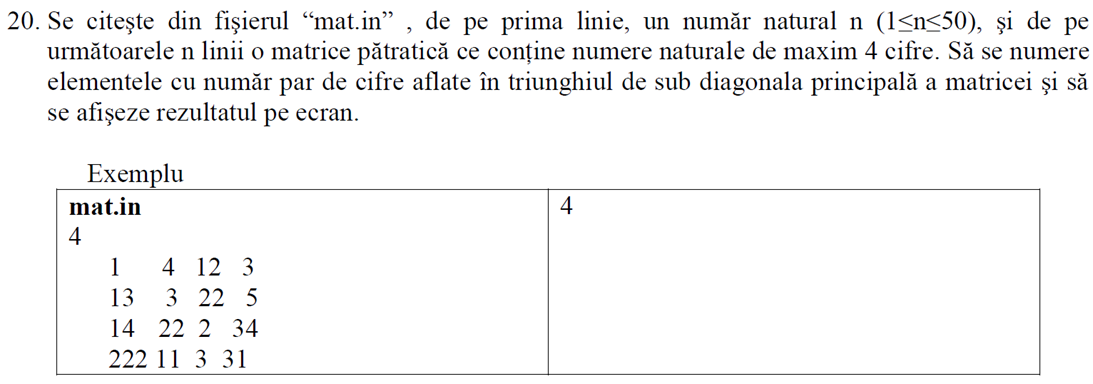
**s=s+a[i][j];**

**}**

**cout<<s;**

**return 0;**

**}**

****

**#include <iostream>**

**#include <fstream>**

**using namespace std;**

**ifstream fin("mat.in");**

**int nrcifre (int x)**

**{ int cifre=0;**

**while (x!=0)**

**{ cifre++;**

**x=x/10;**

**}**

**return cifre;}**

**int main ()**

**{ int a[51][51],n,s=0;**

**fin>>n;**

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

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

**fin>>a[i][j];**

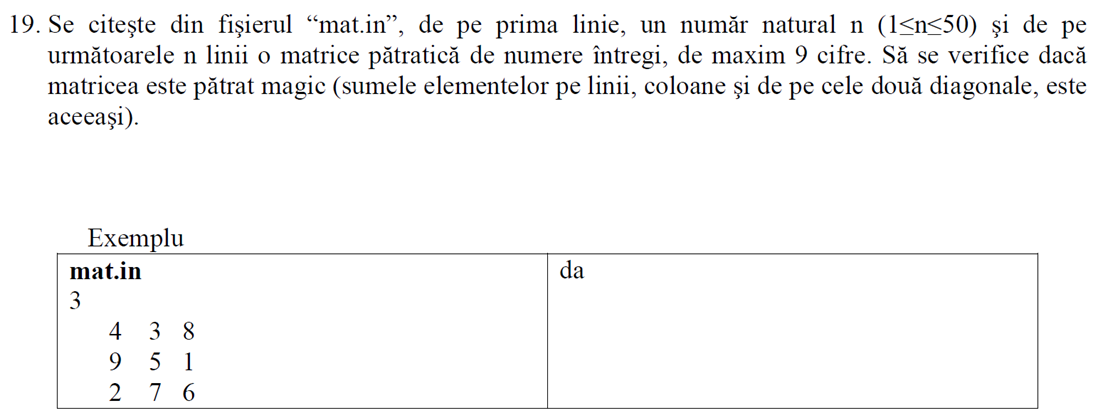
**for (int i=1;i<=n;i++)**

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

**{ if (i>j)  
if (nrcifre(a[i][j]) %2==0) s++;}**

**cout<<s;**

**return 0;}**

****

**#include <iostream>**

**#include <fstream>**

**using namespace std;**

**ifstream fin("mat.in");**

**int main ()**

**{ int a[51][51],n,sdp=0, sds=0,ok=1;**

**fin>>n;**

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

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

**fin>>a[i][j];**

**for(int i=1;i<=n;i++) //calculez suma pe diag. princip ca suma etalon**

**sdp=sdp+a[i][i];**

**for(int i=1;i<=n;i++) //calculez suma pe diag. sec**

**sds=sds+a[i][n+1-i];**

**if(sds!=sdp) ok=0;**

**//calculez sumele de pe fiecare linie**

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

**{ int sl=0;**

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

**sl=sl+a[i][j];**

**if(sl!=sdp)**

**{ok=0;**

**break;}**

**}**

**//calculez sumele de pe fiecare coloana**

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

**{ int sc=0;**

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

**sc=sc+a[i][j];**

**if(sc!=sdp)**

**{ok=0;**

**break;}**

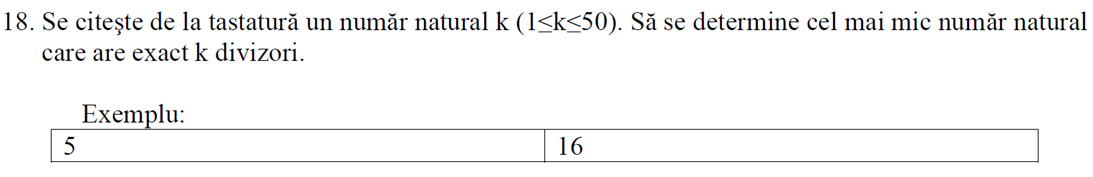
**}**

**if(ok==1)**

**cout<<”DA”;**

**else cout<<”NU”;**

**}**

****

**#include <iostream>**

**using namespace std;**

**int div(int n)**

**{**

**int nd = 0;**

**int i;**

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

**if (n % i == 0) nd += 2;**

**if (i \* i == n)**

**nd++;**

**return nd;**

**}**

**int main()**

**{**

**int k;**

**cin >> k;**

**int ok = 0;**

**int nr = 1;**

**while (!ok)**

**{**

**if (div(nr) == k)**

**{**

**cout << nr;**

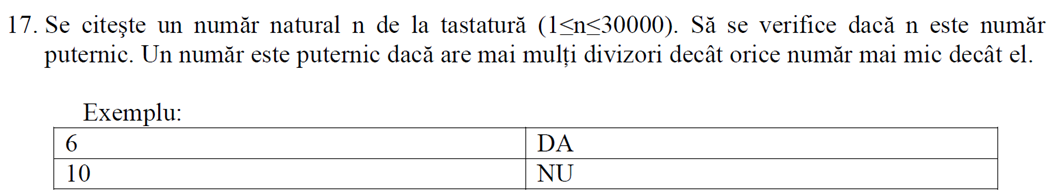
**ok = 1;**

**}**

**nr++;**

**}**

**}**

****

**#include <iostream>**

**using namespace std;**

**int divizori(int n)**

**{**

**int d;**

**int nrd=0;**

**for(d=1; d\*d<n; d++)**

**if(n%d==0)**

**nrd+=2;**

**if(d\*d==n)**

**nrd=nrd+1;**

**return nrd;**

**}**

**int main()**

**{**

**int n,ok=1;**

**cin>>n;**

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

**{**

**if(divizori(i)>=divizori(n))**

**{ok=0;**

**break;}**

**}**

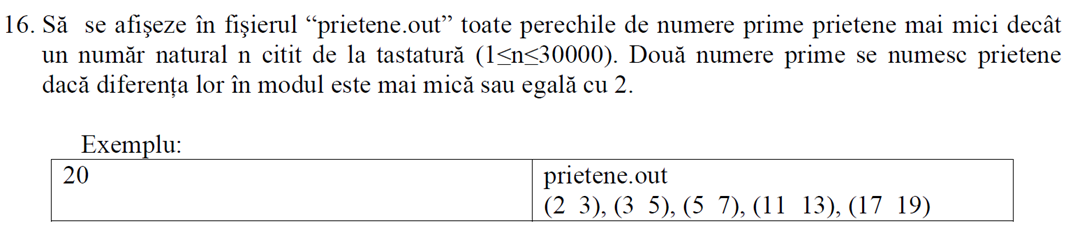
**if(ok==1)**

**cout<<"da";**

**else**

**cout<<"nu";**

**}**

****

**#include <iostream>**

**#include <cmath>**

**#include <fstream>**

**using namespace std;**

**int prim(int n)**

**{**

**if(n<2) return 0;**

**if(n%2==0 && n!=2) return 0;**

**for (int i=3; i\*i<=n; i+=2)**

**if(n%i==0) return 0;**

**return 1;**

**}**

**int main()**

**{**

**int n;**

**cout<<"n=";**

**cin>>n;**

**ofstream f("prietene.out");**

**if(n>=3) f<<2<<" "<<3<<endl;**

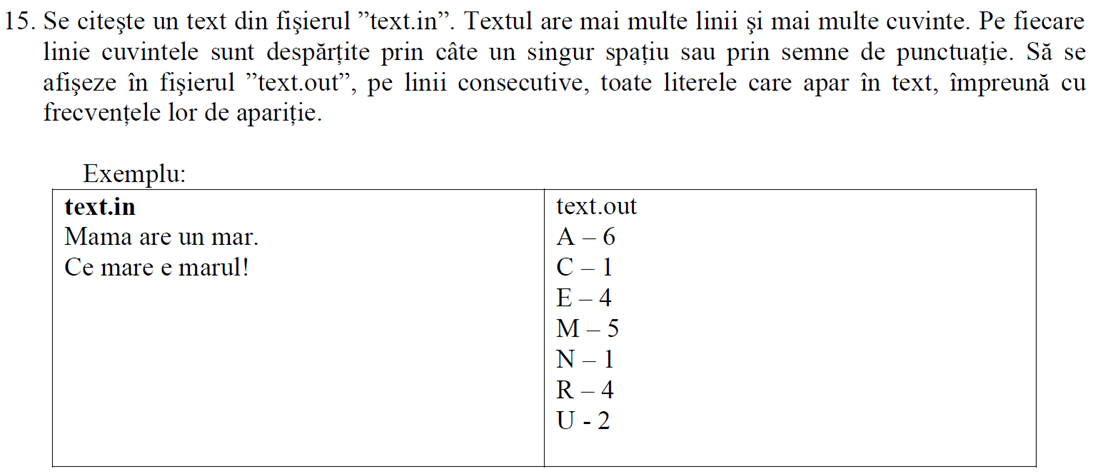
**for(int i=3; i<n-2; i++)**

**if(prim(i) && prim(i+2))**

**f<<i<<" "<<i+2<<endl;**

**return 0;**

**}**

****

**VARIANTA 1**

**#include <iostream>**

**#include <fstream>**

**using namespace std;**

**int main()**

**{**

**char str[256];**

**ifstream fin("bac.txt");**

**ofstream fout("out.txt");**

**int fr[26] = { 0 };**

**while (fin.getline(str, 256))**

**{**

**for (int i = 0; i < strlen(str); i++)**

**{**

**str[i] = (char)toupper(str[i]);**

**if (str[i] >= 'A' && str[i] <= 'Z')**

**fr[str[i] - 'A']++;**

**}**

**}**

**for (int i = 0; i < 26; i++)**

**{**

**if (fr[i] != 0)**

**{**

**cout << (char)(i + 'A') << " " << fr[i] << endl;**

**}**

**}**

**}**

**VARIANTA 2**

**#include<fstream>**

**#include<ctype.h>**

**using namespace std;**

**int nrap[256];**

**int main()**

**{**

**ifstream fin("text.in");**

**char c;**

**while(fin>>c)**

**nrap[toupper(c)]++;**

**fin.close();**

**ofstream fout("text.out");**

**for(c='A'; c<='Z'; c++)**

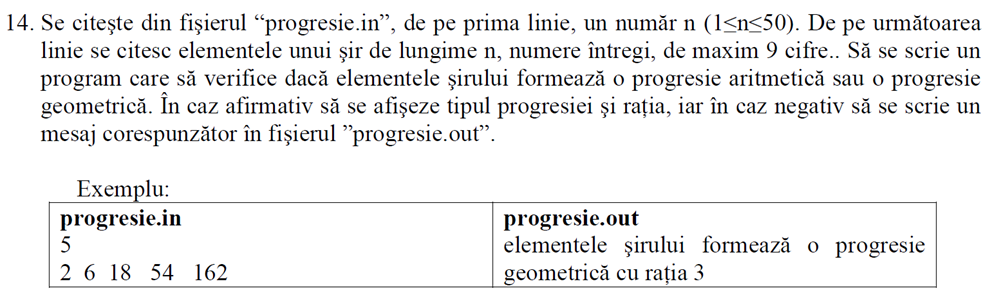
**if(nrap[c]!=0)**

**fout<<c<<" - "<<nrap[c]<<"\n";**

**fout.close();**

**return 0;**

**}**

****

**#include <iostream>**

**#include <fstream>**

**#include <cmath>**

**using namespace std;**

**ifstream f("progresie.in");**

**int main()**

**{**

**int n,a,b,c,r,r1;**

**f>>n;**

**f>>a>>b;**

**int k=1,p=1;**

**while(f>>c)**

**{**

**if((a+c)/2!=b) k=0;**

**else r=b-a;**

**if(a\*c!=b\*b) p=0;**

**else r1=b/a;**

**a=b;**

**b=c;**

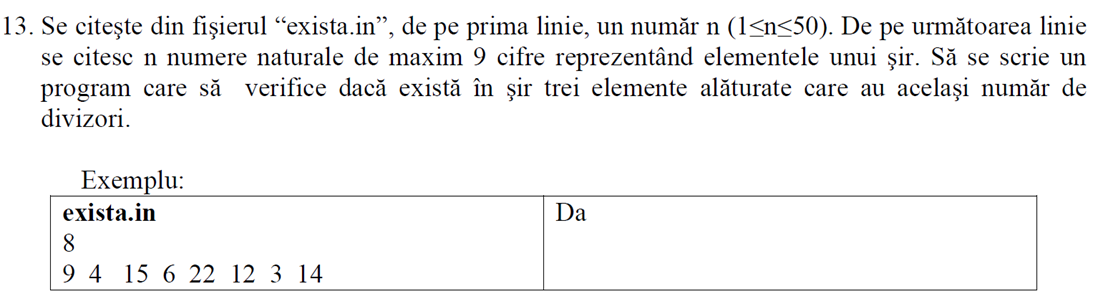
**}**

**if(k==1) cout<<"progresie aritmetica cu ratia "<<r;**

**else if(p==1) cout<<"progresie geometrica cu ratia "<<r1;**

**else cout<<"nu exista";**

**}**

****

**#include <iostream>**

**#include<fstream>**

**using namespace std;**

**ifstream f("exista.in");**

**int nrdiv(int x)**

**{**

**int nrdivizor=0,d;**

**for(d=1;d\*d<x;d++)**

**if(x%d==0)**

**nrdivizor+=2;**

**if(d\*d==x)**

**nrdivizor++;**

**return nrdivizor;**

**}**

**int main()**

**{**

**int n,a,b,c,ok=0;**

**f>>n;**

**f>>a>>b;**

**for(int i=2;i<n;i++)**

**{**

**f>>c;**

**if(nrdiv(a)==nrdiv(b) && nrdiv(b)==nrdiv(c))**

**ok=1;**

**a=b;**

**b=c;**

**}**

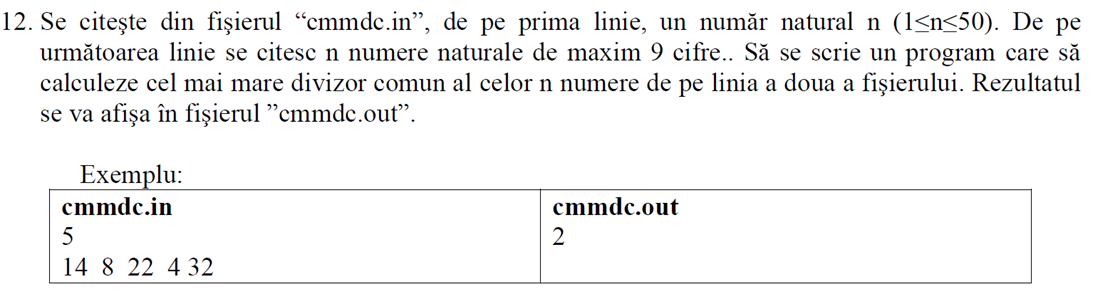
**if(ok==1)**

**cout<<"Da";**

**else cout<<"Nu";**

**return 0;**

**}**

****

**#include <iostream>**

**#include <fstream>**

**using namespace std;**

**int cmmdc(int a, int b)**

**{**

**int r;**

**while (b)**

**{**

**r = a % b;**

**a = b;**

**b = r;**

**}**

**return a;**

**}**

**int main()**

**{**

**int n, a, b;**

**ifstream fin("bac.txt");**

**fin >> n;**

**fin >> a;**

**fin >> b;**

**int cm = cmmdc(a, b);**

**for (int i = 2; i < n; i++)**

**{**

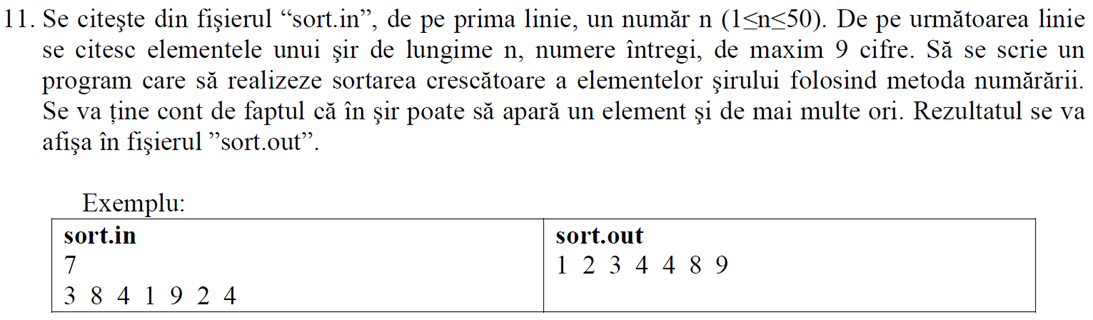
**fin >> b;**

**cm = cmmdc(cm, b);**

**}**

**cout << cm << " ";**

**}**

****

**#include<fstream>**

**using namespace std;**

**int main()**

**{**

**int n,i,j;**

**long a[50], nr[50],b[50];**

**ifstream f("sort.in");**

**ofstream g("sort.out");**

**f>>n;**

**for (i=0;i<n;i++) f>>a[i];**

**for (i=0;i<n;i++) nr[i]=0;**

**for (i=0;i<n;i++)**

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

**if(a[i ]>a[j])**

**nr[i]++;**

**else**

**nr[j]++;**

**for(i=0;i<n;i++) b[nr[i]]=a[i];**

**for (i=0;i<n;i++) g<<b[i]<<" ";**

**f.close();**

**g.close();**

**return 0; }**

***SAU***

**/\***

**Sortarea prin metoda numararii (aia in care nu putem face shir de numarare)**

**- pentru fiecare element a[i] numaram (trecind prin tot shirul initzial) cite elem. is mai mici strict decit el, shi indicele la care va veni in shirul final este dat de indicele ala -1.**

**Pentru alea care se repeta, bagam initzial in shir o valoare care nu e admisa printre datele de intrare**

**\*/**

**#include<fstream>**

**int main()**

**{**

**int a[51],b[51],n,i,j,nr;**

**ifstream fin("sort2.in");**

**fin>>n;**

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

**{**

**fin>>a[i];**

**b[i]=-1000000000;**

**}**

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

**{**

**//numaram kte is strict mai mici dekt a[i]**

**nr=1;**

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

**if(a[j]<a[i]) nr++;**

**b[nr]=a[i];**

**}**

**//umplem golurile:**

**for(i=2;i<=n;i++)**

**if(b[i]==-1000000000)**

**b[i]=b[i-1];**

**fin.close();**

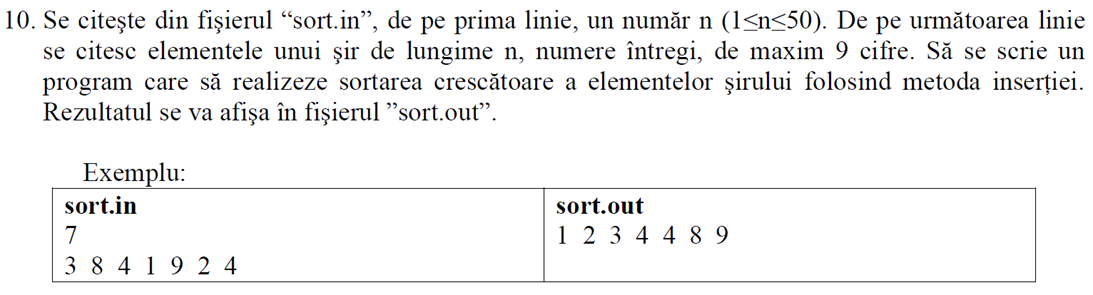
**ofstream fout("sort2.out");**

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

**fout<<b[i]<<" ";**

**fout.close();**

**}**

****

[**https://www.pbinfo.ro/articole/5609/sortarea-prin-insertie**](https://www.pbinfo.ro/articole/5609/sortarea-prin-insertie)

**Sortarea unui şir prin metoda inserţiei (fiecare valoare citita o inseram direct la indicele la care se potriveste (adica la indicele primului elem. mai mare ca ea)**

**#include<fstream.h>**

**void main()**

**{**

**long a[51],n,i,j,k,v;**

**ifstream fin("sort1.in");**

**fin>>n;**

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

**{**

**fin>>v;**

**//cautam v in shirul a, intre indicii elem. deja puse acolo, deci intre indicii 1 shi i-1**

**k=1;**

**while(k<=i-1 && v>=a[k]) k++;**

**//inseram v la indicele k:**

**for(j=i-1;j>=k;j--) a[j+1]=a[j];**

**a[k]=v;**

**}**

**fin.close();**

**ofstream fout("sort1.out");**

**for(i=1;i<=n;i++) fout<<a[i]<<" ";**

**fout.close();**

**}**

**SAU**

**int n, X[100];**

**//citire X[] cu n elemente**

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

**{**

**int x = a[i];**

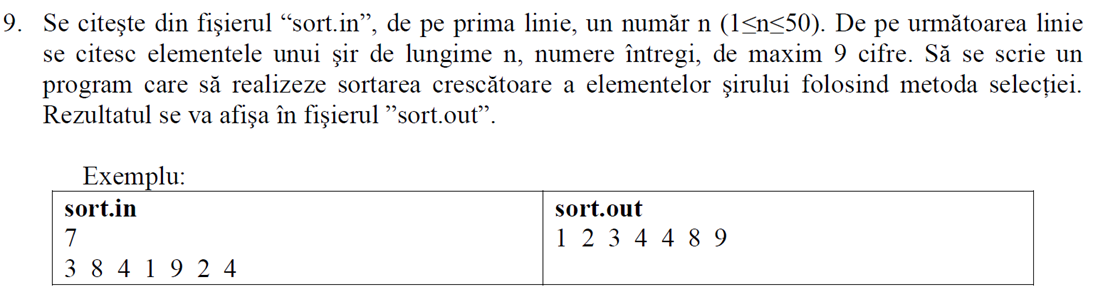
**int p = i - 1;**

**while(p >= 0 && a[p] > x)**

**a[p + 1] = a[p], p --;**

**a[p + 1] = x;**

**}**

****

[**https://www.pbinfo.ro/articole/5605/sortarea-prin-selectie**](https://www.pbinfo.ro/articole/5605/sortarea-prin-selectie)

**/\***

**Sortarea prin selectzie:**

**la fiecare pas k=1, n-1 se determina minimul dintre elem. k..n shi se intersch. cu elem. de la indicele k**

**\*/**

**#include<fstream.h>**

**int main()**

**{**

**long a[51], n, i, k, imin, min, aux;**

**ifstream fin("sort.in");**

**fin>>n;**

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

**fin>>a[i];**

**fin.close();**

**ofstream fout("sort.out");**

**for(k=1;k<=n-1;k++)**

**{**

**min=a[k]; imin=k;**

**for(i=k; i<=n;i++)**

**if(a[i]<min)**

**{**

**min=a[i];**

**imin=i;**

**}**

**aux=a[k];**

**a[k]=a[imin];**

**a[imin]=aux;**

**}**

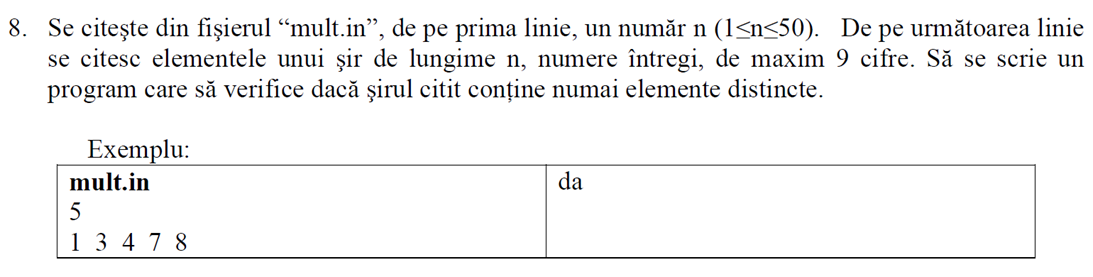
**for(i=1;i<=n;i++) fout<<a[i]<<" ";**

**fout.close();**

**}**

**return 0;**

**}**

****

**int main()**

**{**

**int ok=1, n, a[50];**

**ifstream f("mult.in");**

**f>>n;**

**for (int i=0;i<n;i++)**

**f>>a[i];**

**for (int i=0;i<n-1;i++)**

**for(int j=i+1;j<n;j++)**

**{**

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

**{**

**ok=0;**

**break;**

**}**

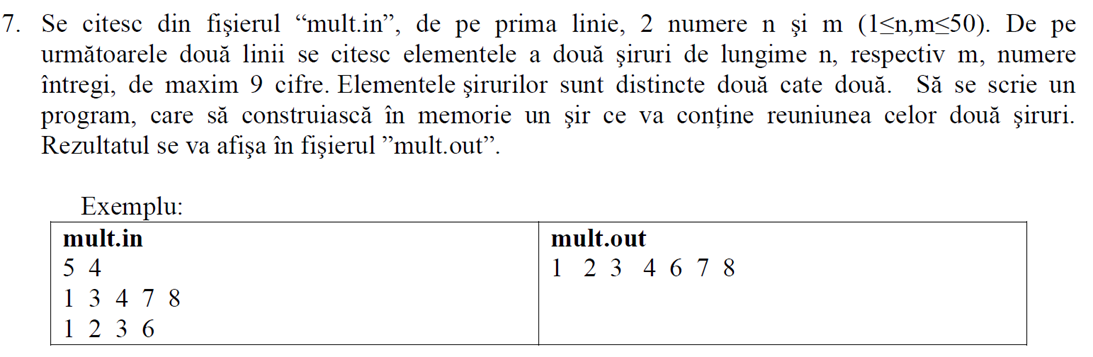
**}**

**if(ok==1) cout<<"da";**

**else cout<<"nu";**

**return 0;**

**}**

****

**#include <iostream>**

**#include <fstream>**

**using namespace std;**

**ifstream d("mult.in");**

**ofstream f("mult.out");**

**int n,m,a[51],b[51],c[102],k;**

**void citire()**

**{int i,j;**

**d>>n;**

**d>>m;**

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

**d>>a[i];**

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

**d>>b[j];**

**}**

**int cautare(int x)**

**{ int i;**

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

**if(c[i]==x)**

**return 1;**

**return 0;**

**}**

**void reuniune ()**

**{ int j,i;**

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

**c[j]=b[j];**

**k=m;**

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

**if(cautare(a[i])==0)**

**c[++k]=a[i];**

**}**

**void afisare()**

**{ int t;**

**for(t=1;t<=k;t++)**

**f<<c[t]<<" ";**

**}**

**int main()**

**{**

**citire();reuniune();afisare();**

**return 0;**

**}**

**VARIANTA INTERCLASARE**

**#include <iostream>**

**#include <fstream>**

**using namespace std;**

**ifstream d("mult.in");**

**ofstream f("mult.out");**

**int n,m,a[51],b[51],c[102],k=0;**

**void citire()**

**{**

**int i,j;**

**d>>n;**

**d>>m;**

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

**d>>a[i];**

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

**d>>b[j];**

**}**

**void interclasare()**

**{**

**int x=0,y=0;**

**while(x<=n && y<=m)**

**{**

**if(a[x]<b[y])**

**{**

**c[k++]=a[x];**

**x++;**

**}**

**else if(a[x]>b[y])**

**{**

**c[k++]=b[y];**

**y++;**

**}**

**else**

**{**

**c[k++]=a[x];**

**x++;**

**y++;**

**}**

**}**

**while(x<=n)**

**c[k++]=a[x++];**

**while(y<=m)**

**c[k++]=b[y++];**

**}**

**void afisare()**

**{ int t;**

**for(t=1;t<k;t++)**

**f<<c[t]<<" ";**

**}**

**int main()**

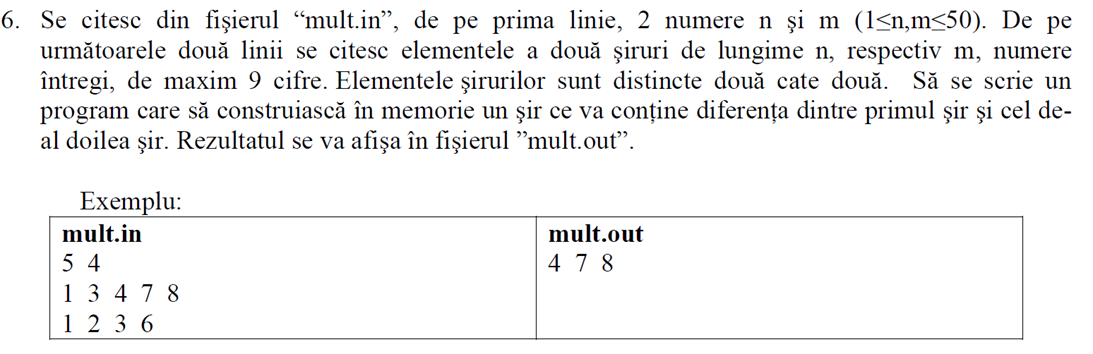
**{**

**citire();**

**interclasare();**

**afisare();**

**}**

****

**#include <iostream>**

**#include <fstream>**

**using namespace std;**

**ifstream d("mult.in");**

**ofstream f("mult.out");**

**int n,m,a[51],b[51],c[102],k=0;**

**void citire()**

**{**

**int i,j;**

**d>>n;**

**d>>m;**

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

**d>>a[i];**

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

**d>>b[j];**

**}**

**int cautare(int x)**

**{**

**int i;**

**for(i=1; i<=m; i++)**

**if(b[i]==x)**

**return 1;**

**return 0;**

**}**

**void diferenta ()**

**{**

**for (int i=1;i<=m;i++)**

**if(cautare(a[i])==0)**

**c[++k]=a[i];**

**}**

**void afisare()**

**{**

**for(int t=1;t<=k;t++)**

**f<<c[t]<<" ";**

**}**

**int main ()**

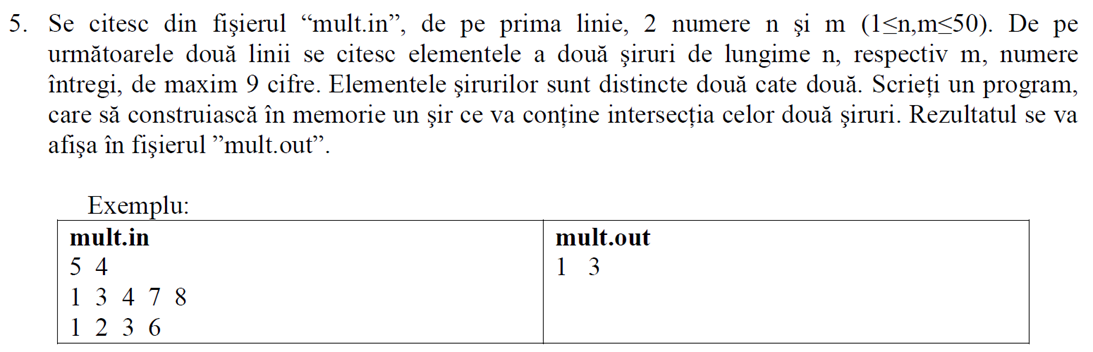
**{**

**citire();**

**diferenta();**

**afisare();**

**}**

****

**#include <iostream>**

**#include <fstream>**

**using namespace std;**

**ifstream d("mult.in");**

**ofstream f("mult.out");**

**int n,m,a[51],b[51],c[102],k=0;**

**void citire()**

**{**

**int i,j;**

**d>>n;**

**d>>m;**

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

**d>>a[i];**

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

**d>>b[j];**

**}**

**void intersectie ()**

**{**

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

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

**{**

**if(a[i]==b[j])**

**c[++k]=a[i];**

**}**

**}**

**void afisare()**

**{**

**for(int t=1;t<=k;t++)**

**f<<c[t]<<" ";**

**}**

**int main ()**

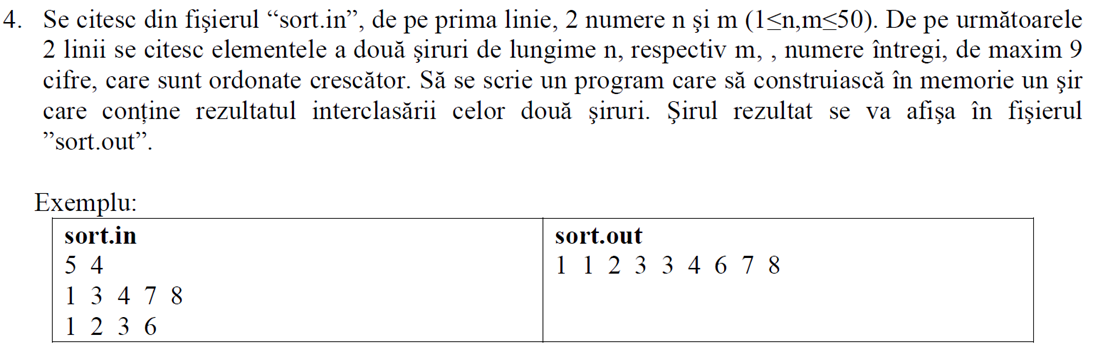
**{**

**citire();**

**intersectie();**

**afisare();**

**}**

****

**#include <iostream>**

**#include <fstream>**

**using namespace std;**

**ifstream d("mult.in");**

**ofstream f("mult.out");**

**int n,m,a[51],b[51],c[102],k=1;**

**void citire()**

**{**

**int i,j;**

**d>>n;**

**d>>m;**

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

**d>>a[i];**

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

**d>>b[j];**

**}**

**void interclasare()**

**{**

**int x=1,y=1;**

**while(x<=n && y<=m)**

**{**

**if(a[x]<b[y])**

**{**

**c[k++]=a[x];**

**x++;**

**}**

**else**

**{**

**c[k++]=b[y];**

**y++;**

**}**

**}**

**while(x<=n)**

**c[k++]=a[x++];**

**while(y<=m)**

**c[k++]=b[y++];**

**}**

**void afisare()**

**{ int t;**

**for(t=1;t<k;t++)**

**f<<c[t]<<" ";**

**}**

**int main()**

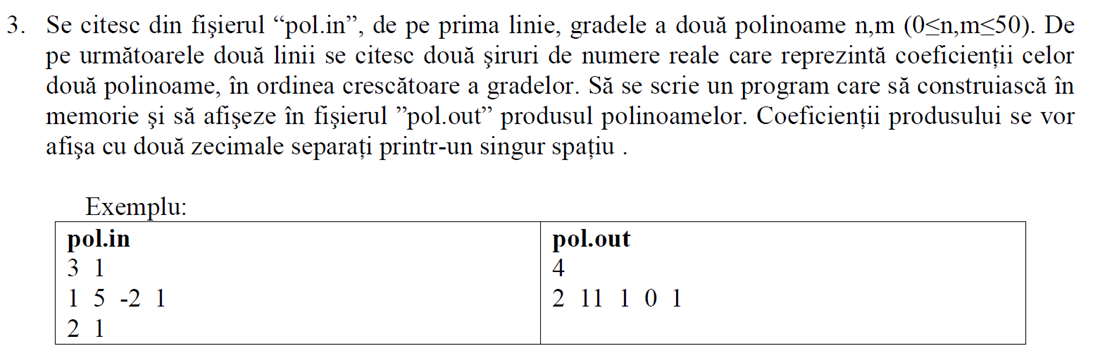
**{**

**citire();**

**interclasare();**

**afisare();**

**}**

****

**p1=1+5x-2x^2+x^3**

**p2=2+x 0 1 2 3 4**

**p1\*p2=(1+5x-2x^2+x^3)\*(2+x)=1\*2+x\*(1\*1+5\*2)+x^2\*(-2\*2+5\*1)+x^3(2-2\*1)+x^4\*1**

**#include <iostream>**

**#include <fstream>**

**using namespace std;**

**int main()**

**{**

**ifstream f ("pol.in");**

**ofstream g ("pol.out");**

**float p1[55], p2[55], pr[55]={0};**

**int m,n,grad;**

**f>>n>>m;**

**for(int i=0; i<=n; i++)**

**f>>p1[i];**

**for(int i=0; i<=m; i++)**

**f>>p2[i];**

**grad=m+n;**

**for(int i=0; i<=n; i++)**

**for(int j=0; j<=m; j++)**

**{**

**pr[i+j]=pr[i+j]+p1[i]\*p2[j];**

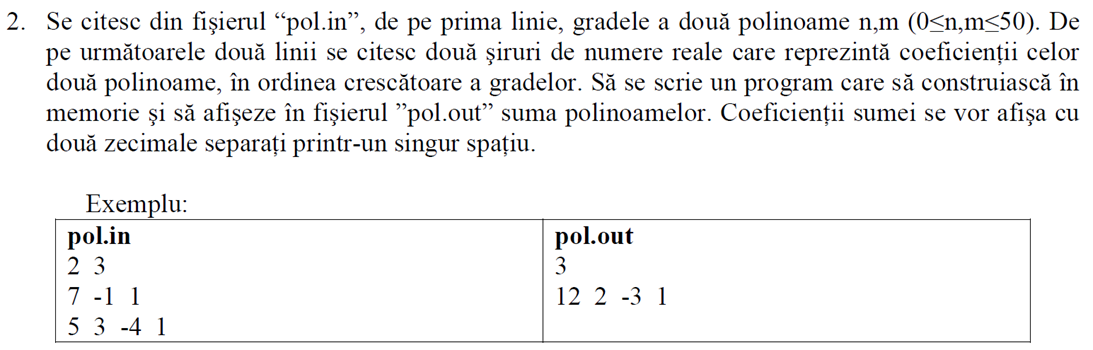
**}**

**g<<grad<<endl;**

**for(int i=0; i<=grad; i++)**

**g<<pr[i]<<" ";**

**}**

****

**p1=7-x+x^2**

**p2=5+3x-4\*x^2+x^3**

**suma=p1+p2 =(5+7)+x\*(-1+3)+x^2(1-4)+x^3\*(0+1)**

**#include <iostream>**

**#include <fstream>**

**using namespace std;**

**int main()**

**{**

**ifstream f ("pol.in");**

**ofstream g ("pol.out");**

**float p1[55], p2[55], suma[55];**

**int m,n,grad;**

**f>>n>>m;**

**for(int i=0; i<=n; i++)**

**f>>p1[i];**

**for(int i=0; i<=m; i++)**

**f>>p2[i];**

**if(n>m)**

**grad=n;**

**else grad=m;**

**for(int i=0; i<=grad; i++)**

**{ suma[i]=p1[i]+p2[i];**

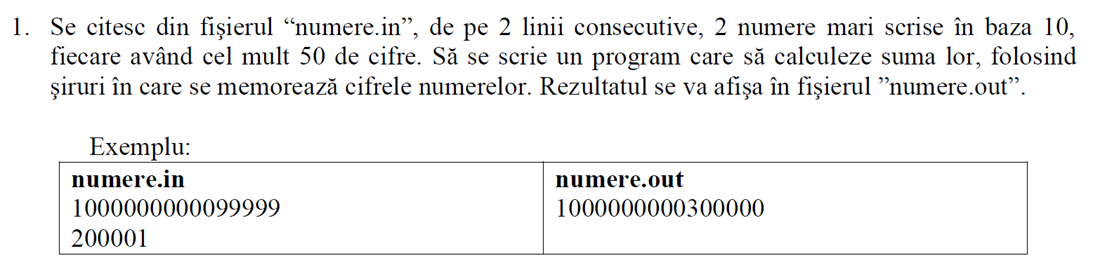
**}**

**g<<grad<<endl;**

**for(int i=0; i<=grad; i++)**

**g<<suma[i]<<" ";**

**}**

****

**int main()**

**{**

**ifstream fin("numere.in");**

**char s[100];**

**int i,t,a[51],b[51],c[51],na,nb,nc;**

**fin.getline(s,100);**

**strrev(s);**

**na=strlen(s);**

**for(i=0;i<na;i++)**

**a[i]=s[i]-'0';**

**fin.getline(s,100);**

**strrev(s);**

**nb=strlen(s);**

**for(i=0;i<nb;i++)**

**b[i]=s[i]-'0';**

**nc=max(na,nb);//am folosit functzia predefinita max**

**//completam cu 0-uri vectorashul mai mic:**

**for(i=na;i<nc;i++)a[i]=0;//obs: din astea doua for-uri se**

**for(i=nb;i<nc;i++)b[i]=0;// executa cel mult unul.**

**t=0;**

**for(i=0;i<nc;i++)**

**{**

**c[i]=(a[i]+b[i]+t)%10;**

**t=(a[i]+b[i]+t)/10;**

**}**

**if(t)**

**c[nc++]=t;**

**ofstream fout("numere.out");**

**for(i=nc-1;i>=0;i--)**

**fout<<c[i];**

**fout.close();**

**return 0;**

**}**