

**Probleme cu vectori**

Sa se afiseze toate numerele din intervalul [1000,99999] care au proprietatea ca atat ele cat si rasturnatul lor sunt prime.

```
#include<iostream>
#include<math.h>
long n,c,d,k;
int prim(long n)
{if(n==0 || n==1) return 0;
  for(int i=2;i<=sqrt(n);i++)
    if(n%i==0) return 0;
  return 1;
}
long rast(long n)
{d=0;
  while(n!=0)
  {c=n%10;
    d=d*10+c;
    n=n/10;
  }
  return d;
}
void main()
{for(long int i=1001;i<=99999;i++)
  if(prim(i) && prim(rast(i))) cout<<i<<endl;
}
```

Se citeste un vector A cu n elemente naturale. Sa se afiseze pentru fiecare element din vector suma divizorilor sai proprii.

```
#include<fstream.h>
ifstream f("sumadiv.in");
int n, a[100];
void citire()
{ int i;
  f>>n;
  for(i=1;i<=n;i++)
    f>>a[i];
}
int sumadiv(int n)
{
  int i,suma=0;
  for(i=2;i<n;i++)
    if(n%i==0)
      suma=suma+i;
  return suma;
}
int main()
{
  citire();
  int i;
```

```

for(i=1;i<=n;i++)
if(a[i]<sumadiv(a[i]))
cout<<a[i]<<endl;
f.close();
return 0;
}

```

Se citesc 4 numere naturale. Sa se calculeze cmmdc al lor.

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

```

int cmmdc(int x, int y)
{ while(x!=y)
  if(x>y) x=x-y;
  else y=y-x;
  return x;
}

```

```

void main()
{ int a,b,c,d;
  cin>>a>>b>>c>>d;
  cout<<cmmdc(cmmdc(a,b),cmmdc(c,d));
}

```

Se citeste un vector cu n elemente numere naturale. Sa se calculeze suma cifrelor din care sunt compuse elementele vectorului.

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

```

int sumacif(int n)
{ int sc=0;
  while(n)
  { sc=sc+n%10;
    n=n/10;
  }
  return sc;
}

```

```

void main()
{ int a[100], n, i, s=0;
  cin>>n;
  for(i=1;i<=n;i++) cin>>a[i];
  for(i=1;i<=n;i++) s=s+sumacif(a[i]);
  cout<<s;
}

```

Se citeste un vector cu n elemente numere naturale. Sa se ordoneze descrescator dupa valoarea rasturnatului.

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

```

void inter(int &x, int &y)
{ int aux=x; x=y; y=aux;
}

```

```
int rast(int n)
```

```

{ int r=0;
  while(n)
  { r=r*10+n%10;
    n=n/10;
  }
  return r;
}

void sort(int a[100], int n)
{ int i,gata;
  do { gata=1;
    for(i=1;i<n;i++)
      if(rast(a[i])<rast(a[i+1]))
        { inter(a[i],a[i+1]);
          gata=0;
        }
  }
  while(!gata);
}

void main()
{ int a[100],n,i;
  cin>>n;
  for(i=1;i<=n;i++) cin>>a[i];
  sort(a,n);
  for(i=1;i<=n;i++) cout<<a[i]<<" ";
}

```

Se citeste un vector cu n elemente numere naturale. Sa se ordoneze descrescator dupa valoarea rasturnatului.

```

#include<iostream.h>
void inter(int &x, int &y)
{ int aux=x; x=y; y=aux;
}

int rast(int n)
{ int r=0;
  while(n)
  { r=r*10+n%10;
    n=n/10;
  }
  return r;
}

void sort(int a[100], int n)
{ int i,gata;
  do { gata=1;
    for(i=1;i<n;i++)
      if(rast(a[i])<rast(a[i+1]))
        { inter(a[i],a[i+1]);

```

```

        gata=0;
    }
}
while(!gata);
}
void main()
{ int a[100],n,i;
  cin>>n;
  for(i=1;i<=n;i++) cin>>a[i];
  sort(a,n);
  for(i=1;i<=n;i++) cout<<a[i]<<" ";
}

```

Sa se scrie o functie care primeste 3 parametri: n - numar natural, c1,c2 cifre si returneaza numarul obtinut din n prin inlocuirea tuturor aparitiilor cifrei c1 cu c2.

Ex. din n=2324, c1=2 si c2=5 returneaza 5354

```
#include<iostream >
```

```

int schimba(long n, int c1, int c2)
{ long r=0;
  while(n)
  { if(n%10==c1)
    r=r*10+c2;
    else r=r*10+n%10;
    n=n/10;
  }
  while(r)
  { n=n*10+r%10;
    r=r/10;
  }
  return n;
}

```

```

void main()
{ cout<<schimba(2324,2,5);
}

```

Sa se scrie o functie interval care primeste 3 parametri n, a, b numere intregi. Functia determina un interval (a,b) de lungime minima cu proprietatea ca n apartine intervalului (a,b) si ca a si b sunt numere prime. Capetele intervalului vor fi returnate prin intermediul parametrilor a si b.

```
#include<iostream >
```

```

int prim (int n)
{ if(n==0 || n==1) return 1;
  for(int i=2;i<=n/2;i++)
    if(n%i==0) return 0;
  return 1;
}

```

```

void interval(int n, int &a, int &b)
{
    a=n-1;
    while(!prim(a)) a--;
    b=n+1;
    while(!prim(b)) b++;
}

```

```

void main()
{
    int n,a,b;
    cin>>n;
    interval(n,a,b);
    cout<<a<<" "<<b;
}

```

Sa se afiseze toate numerele din intervalul [100,99999] care au proprietatea sunt prime si sunt palindroame. Un numar este palindrom daca citit de la dreapta la stanga are aceeasi valoare.

```

using namespace std;
#include<iostream>
#include<math.h>
long n,c,d,k;
int prim(long n)
{
    if(n==0 || n==1) return 0;
    for(int i=2;i<=sqrt(n);i++)
        if(n%i==0) return 0;
    return 1;
}
long rast(long n)
{
    d=0;
    while(n!=0)
    {
        c=n%10;
        d=d*10+c;
        n=n/10;
    }
    return d;
}
int main()
{
    for(long int i=100;i<=1000;i++)
        if(prim(i) && rast(i)==i) cout<<i<<endl;
}

```

Sa se scrie o functie care primeste ca parametru un numar natural n si afiseaza descompunerea lui n in factori primi.

Ex.  $n=45 \Rightarrow 3 \cdot 3 \cdot 5$

```
#include<iostream >
```

```

void desc(int n)
{
    int d=2;
    while(n>1)

```

```

    if(n%d==0) { cout<<d<<" ";
                n=n/d;
            }
    else d++;
}

```

```

void main()
{ desc(45);
}

```

Sa se scrie o functie care primeste ca parametru un numar natural n si returneaza suma exponentilor din descompunerea in factori primi a lui n.

```

#include<iostream >
int exponenti(int n)
{ int d=2,x=0;
  while(n>1)
  { if(n%d==0) { x++;
                n=n/d;
            }
    else d++;
  }
  return x;
}

```

```

void main()
{ cout<<exponenti(45);
}

```

Sa se scrie o functie care primeste ca parametru un numar natural n si returneaza cea mai mare putere a lui 2 mai mica sau egala cu n. Folosind aceasta functie sa se decompuna un numar ca suma de puteri ale lui 2.

```

#include<iostream>
using namespace std;

```

```

int Putere( int n )
{
    int x = 1;
    while ( x <= n )
        x = x * 2 ;
    return x/2;
}

```

```

int main()
{
    int n;
    cin >> n;
    while ( n != 0 )
    {
        cout << Putere(n) << ' ';
        n = n - Putere(n);
    }
}

```

```

    }

    system("pause");
return 0;
}

```

Sa se scrie o functie ordonare care primeste 3 parametri: un vector a cu maxim 100 de elemente numere intregi, un numar natural n reprezentand numarul efectiv de elemente din vectorul a si un numar natural k mai mic decat n. Functia ordoneaza crescator primele k elemente ale vectorului a si descrescator celelalte n-k.

```

#include<iostream>
using namespace std;

int Ordonare( int a[100] , int n , int k)
{
    int i;
    int j;
    int aux = 0;
    for ( i = 1 ; i < k ; i++ )
        for ( j = i + 1 ; j <= k ; j++ )
            if ( a[i] > a[j] )
            {
                aux = a[i];
                a[i] = a[j];
                a[j] = aux;
            }

    for ( i = k + 1 ; i < n ; i++ )
        for ( j = i + 1 ; j <= n ; j++ )
            if ( a[i] < a[j] )
            {
                aux = a[i];
                a[i] = a[j];
                a[j] = aux;
            }

}

int main ()
{
    int a[100];
    int n;
    int k;
    cin >> n;
    cin >> k;
    for ( int i = 1 ; i <= n ; i++ )
        cin >> a[i];
    Ordonare( a , n , k );
    for ( int i = 1 ; i <= n ; i++ )
        cout<< a[i] << ' ';
    system("pause");
}

```

```

    return 0;
}

```

Sa se scrie o functie Sufix care primeste ca parametru un numar natural x si determina daca x este sufix al patratului sau (de exemplu 25 este sufix al lui 625). Folosind aceasta functie sa de determine numerele naturale din intervalul [1,n] care sunt sufixe ale patratelor lor.

```
#include<iostream>
```

```
using namespace std;
```

```

int Sufix( int x )
{
    long a = x * x;

    while ( x != 0 )
        if ( x % 10 != a %10 )
            return 0;
        else
        {
            x = x /10;
            a = a /10;
        }
    return 1;
}

```

```

int main()
{
    int n;
    cin >> n;
    for ( int i = 1 ; i < n; i++ )
        if ( Sufix(i) )
            cout << i << ' ';
    system("pause");
    return 0;
}

```

Se numeste cifra de control cifra obtinuta prin calculul reperat al sumei cifrelor unui numar natural. De exemplu 66 are suma de control 3 ( $6+6=12$ ,  $1+2=3$ ).

Se citeste un vector a cu n elemente numere naturale. Sa se afiseze elementele vectorului grupate pe randuri separate in ordine crescatoare a cifrei de control.

```
#include<iostream>
```

```
using namespace std;
```

```

int SumaCf( int a )
{
    int s = 0;
    while ( a != 0 )
    {

```



```

        s = s + a % 10;
        a = a/10;
    }
    return s;
}

int Cifra( int a )
{
    do
        a = SumaCf(a);
    while ( a >= 10 );
    return a;
}

int main()
{
    int a[100];
    int n;
    cin >> n;
    for ( int i = 0 ; i < n; i++)
        cin >> a[i];
    for ( int c = 1 ; c <= 9 ; c++ )
    {
        for ( int i = 0 ; i < n; i++)
            if ( Cifra(a[i]) == c )
                cout << a[i] << ' ';
        cout << "\n";
    }
    system("pause");
    return 0;
}

```

**Scriti o functie care sa calculeze si sa returneze numarul de divizori primi ai unui numar natural n primit ca parametru.**

```

#include<iostream>
int nrdivprimi(int n)
{ int d=2,k=0;
  while(n>1)
      if(n%d==0) { k++;
                  while(n%d==0) n=n/d;
                  }
                  else d++;
  return k;
}

int main()
{
    int n;
    cin>>n;
}

```

```

    cout<<nrdivprimi(n);
    return 0;
}

```

### Fisiere text in C++

Fisier text=sucesiune de caractere ASCII ce se incheie cu caracterul EOF.

Accesul la caracterele scrise intr-un fisier text se poate face doar secvential, si poate fi :

Acces pentru citire

Acces pentru scriere

Toate utilitarele necesare pentru accesarea fisierelor text din programe C++ se afla in biblioteca fstream.h

Pentru a putea accesa un fișier text dintr-un program C++ trebuie declarata o variabila, astfel :

Pentru a avea acces la afisare in fișier : ofstream g;

Pentru a avea acces la citirea din fișier : ifstream f;

Cand ne hotarâm sa accesam fisierul, mai intai trebuie sa-l "deschidem". De fapt, creem legatura logica intre variabila declarata pentru accesarea fisierului si zona de pe disc unde se gaseste fisierul. Dupa crearea acestei legaturi, ori de cate ori dorim sa accesam fisierul vom accesa de fapt variabila asociata lui.

Ex.

```

    f.open("intrare.txt");
    g.open("iesire.txt");
    }

```

cele 2 fisiere se cauta in folderul de lucru

f.open("c:\\teza\\unu.in"); fisierul se cauta pe calea specificata

```

ifstream f("intrare.txt");
ofstream g("iesire.txt");
    }

```

initializare variabila de acces, la declarare

Cand ne hotarâm sa nu mai accesam fisierul va trebui sa-l "inchidem". De fapt, distrugem legatura logica intre variabila declarata pentru accesarea fisierului si zona de pe disc unde se gaseste fisierul.

Ex.

```

f.close();    g.close();

```

Afisarea in fisier

Se face la fel ca si afisarea pe ecran numai ca metoda cout se inlocuieste cu numele variabilei de acces la fisier.

Ex.

```
g<<17<' ';
g<<"ana are mere"<<endl;
```

**Aplicatie :** Sa se afiseze in fisierul text unu.out, toate literele mari, in ordine invers alfabetica si separate prin cate un spatiu.

```
#include <fstream.h>
int main() {
    char L;
    ofstream g("unu.out");
    for (L='Z'; L>='A'; L--)
        g<<L<<' ';
    g.close();
    return 0;
}
```

### Citirea din fisier

Operatorul >> permite citirea datelor in variabila de orice tip, fara insa a trata caracterele albe ( spatii, salturi la linie noua ).  
Citirea se face similar celei de la tastatura numai ca se inlocuieste metoda cin cu variabila de acces la fisier.

**Ex1:** Se considera fisierul text unu.in care contine pe o singura linie si separate prin spatii 10 numere intregi de maxim 4 cifre fiecare. Sa se afiseze pe ecran suma acestor numere.

```
#include <fstream.h>
#include <iostream.h>
int main() {
    int x; long s=0;
    ifstream f("unu.in");
    int i;
    for(i=1;i<=10;i++) {
        f>>x; s=s+x; }
    f.close();
    cout<<s;
    return 0;
}
```

**Ex2:** Se considera fisierul text doi.in care contine: pe prima linie numarul natural nenul  $n$  ( $n \leq 50$ ) iar pe linia a doua, separate prin spatiu,  $n$  numere naturale cu maxim 9 cifre fiecare. Sa se afiseze pe ecran cel mai mare dintre cele  $n$  numere.

```
#include <fstream.h>
#include <iostream.h>
int main() {
    long x, max; int n,i;
```

```

    ifstream f("doi.in");
    f>>n>>max;
    for(i=2;i<=n;i++) {
        f>>x;
        if (x>max) max=x; }
    f.close();
    cout<<max;
return 0;
}

```

Ex3: Se considera fisierul text trei.in care contine pe mai multe linii litere mici in cadrul fiecarei linii fiind separate prin spatiu. Sa se afiseze, in fisierul text trei.out, toate literele continute in unu.in dar scrise cu majuscule si lipite unele de altele.

```

#include <fstream.h>
int main() {
    char L;
    ifstream f("trei.in");
    ofstream g("trei.out");
    while (f>>L)
        g<<(char)(L-32);
    g.close();f.close();
    return 0;
}

```

Explicatie : expresia f>>L va avea valoarea 1 ( adevarat ) daca citirea s-a produs respectiv valoarea 0 ( fals ) daca citirea nu s-a produs – deoarece fisierul s-a terminat.

Ex4: Se considera fisierul text patru.in. care contine: pe prima linie litere separate prin spatiu iar pe linia a doua numere separate prin spatiu. Sa se afiseze in fisierul text patru.out pe prima linie cate litere si pe linia a doua cate numere sunt in patru.in.

```

#include <fstream.h>
int main() {
    char L; long n; int nr=0;
    ifstream f("patru.in"); ofstream g("patru.out");
    while ( f.peek()!='\n') { f>>L; nr++; }
    g<<nr<<endl;
    nr=0;
    while (f>>n) nr++;
    g<<nr;
    g.close(); f.close();
    return 0;
}

```

Explicatie : funcția peek() determina caracterul ce urmeaza in fisier.

Ex5: Se considera fisierul text cinci.in care contine mai multe linii, pe fiecare linie aflandu-se numere naturale cu cel mult 8 cifre fiecare, separate prin spatiu. Cate linii contine fisierul ?

```
#include <fstream.h>
#include <iostream.h>
int main() {
    int nr=0; long x;
    ifstream f("cinci.in");
    while ( f.peek()!=EOF ) {
        f>>x; if ( f.peek()=='\n') nr++;
    }
    nr++;
    f.close();
    cout<<nr;
    return 0;
}
```

**Functia get() permite citirea unui caracter de orice fel (inclusiv spatiu ori salt la linie noua).**

**Ex1: se considera fisierul text unu.in care contine doua linii de text.  
Cate spatii se afla pe fiecare linie ?**

```
#include <fstream.h>
#include <iostream.h>
int main() {
    char c; int nr=0;
    ifstream f("unu.in");
    while ( f.peek()!='\n') {
        c=f.get(); if (c==' ') nr++; }
    f.get(); /* citim enter-ul dar nu il memoram */
    cout<<nr<<' '; nr=0;
    while (f.peek()!=EOF) {
        c=f.get(); if (c==' ') nr++; }
    f.close();
    cout<<nr;
    return 0;
}
```

**Ex2: Se considera fisierul text doi.in care contine mai multe linii. Sa se afiseze cu exactitate continutul fisierului pe ecran.**

```
#include <fstream.h>
#include <iostream.h>
int main() {
    char c;
    ifstream f("doi.in");
    while (f.peek()!=EOF) {
        c=f.get();
        cout<<c; }
    f.close();
    return 0;
}
```

**Ex3:** Se considera fisierul text trei.in care contine mai multe linii. Sa se afiseze, pentru fiecare linie, ce lungime in caractere imprimabile are ?

```
#include <fstream.h>
#include <iostream.h>
int main() {
    int nr=0; int L=0;
    char c;
    ifstream f("trei.in");
    while ( f.peek()!=EOF ) {
        nr++;
        c=f.get();
        if (c=='\n') { L++;
            cout<<L<<' '<<nr-1<<endl;
            nr=0; }
    }
    cout<<L+1<<' '<<nr;
    f.close();
    return 0;
}
```

**Ex4.** Se considera fisierele text unu.txt si doi.txt. Sa se verifice daca cele 2 fisiere sunt identice.

```
#include <fstream.h>
#include <iostream.h>
int main() {
    char c1,c2;
    int ok=1;
    ifstream f("unu.txt");
    ifstream g("doi.txt");
    while (f.peek()!=EOF && g.peek()!=EOF) {
        c1=f.get(); c2=g.get();
        if (c1!=c2) { ok=0; break; }
    }
    ok=ok && (f.peek()==EOF) && (g.peek()==EOF);
    g.close();
    f.close();
    if (ok) cout<<"da";
    else cout<<"nu";
    return 0;
}
```

Functia getline() permite citirea unui sir de caractere, pana la sfarsitul liniei.

**Ex:** Se considera fisierul text unu.in care contine mai multe linii de text. Sa se afiseze pe ecran liniile de text si numarul acestora.

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

```
#include <stdio.h>
int main() {
    char t[1000]; int L=0;
    ifstream f("unu.in");
    while ( f.peek()!=EOF ) {
        L++;
        f.getline(t,1000);
        puts(t);
    }
    f.close();
    cout<<L;
    return 0;
}
```

### Stergerea unui fisier text de pe disc

Exemplu : unlink ("unu.in");

acest fisier nu trebuie sa fie deschis

### Redenumirea unui fisier text de pe disc

Exemplu : rename ("vechi.in", "nou.in"); fisierul nou.in trebuie sa existe deja

deschis  
fisierul vechi.in nu trebuie sa fie

Funcțiile rename si unlink se afla in biblioteca stdio.h

Ex1 Se considera fisierul text unu.in care contine mai multe linii de text. Sa se inlocuiasca in fisier toate caracterele imprimabile cu codul lor ASCII precedat de caracterul #.

Exemplu: daca fisierul unu.in are continutul alaturat  
el va avea continutul de mai jos  
#65#49#98  
#51#67

A1b  
3C

```
#include <stdio.h>
#include <fstream.h>
int main() {
    char c;
    ifstream f("unu.in");
    ofstream g("copie.out");
    while (f.peek()!=EOF) {
        c=f.get();
        if (c!='\n' && c!=' ') g<<'#<<(int)c;
        else g<<c;
    }
    g.close();
    f.close();
}
```

```

        unlink("unu.in");
        rename("copie.out","unu.in");
    return 0;
}

```

Ex2 Fisierul text doi.in contine numere intregi dispuse pe mai multe linii, in cadrul fiecărei linii fiind separate prin spatiu. Se dau de la tastatura numele intregi x, y. Sa se inlocuiasca in fisier toate numerele egale cu x, prin numarul y.

Exemplu: daca fisierul doi.in are continutul alaturat

	12 -13 145 -9 -13
iar x=—13, y=5 fisierul va avea continutul de mai jos	-13
12 5 145 -9 5	890 78
5	
890 78	

```

#include <stdio.h>
#include <fstream.h>
#include <iostream.h>
int main() {
    int x,y,z;
    char c;
    cin>>x>>y;
    ifstream f("doi.in");
    ofstream g("copie.out");
    while (f.peek()!=EOF) {
        f>>z;
        if (z==x) z=y;
        g<<z;
        if (f.peek()!='\n' && f.peek()!=EOF ) g<<' ';
        else g<<endl;
    }
    g.close();f.close();
    unlink("doi.in");
    rename("copie.out","doi.in");
    return 0;
}

```

Sa se scrie o functie care primeste ca parametri un vector x cu maxim 100 de elemente numere naturale si un numar natural n reprezentand numarul de elemente din vector si citeste numarul n si elementele vectorului x.

b) Sa se scrie o functie care primeste ca parametri un vector x cu maxim 100 de elemente numere naturale si un numar natural n reprezentând numarul de elemente din vector. Functia calculeaza si returneaza suma obtinuta adunând ultima cifra din fiecare element al vectorului x.

c) Folosind functiile de mai sus sa se citeasca un vector si sa se afiseze valoarea obtinuta prin adunarea ultimei cifre din fiecare element al vectorului.

```

#include<fstream>
using namespace std;
ifstream fin("date.in");

```



```
ofstream fout("date.out");

void citire(int a[100], int &n)
{
    int i;
    fin>>n;
    for(i=1;i<=n;i++) fin>>a[i];
}

int suma(int a[100], int n)
{
    int i,s=0;
    for(i=1;i<=n;i++) s=s+a[i]%10;
    return s;
}

int main()
{
    int a[100], i,n;
    citire(a,n);
    fout<<suma(a,n);
    fin.close();
    fout.close();
    return 0;
}
```