**Algoritmi fundamentali**

1. **Prelucrarea cifrelor unui numar natural n**

**P1) Suma cifrelor lui n**

Exemplu: n=295

295=29\*10+5, c=5

29=2\*10+9, c=9

2=0\*10+2, c=2

n=0

#include<iostream>

using namespace std;

int n, c, s; // variabile globale

int main()

{ cin>>n;

while (n) // sau (n!=0), sau (n>0)

{ c=n%10; // c= cifra curenta

s+=c; // sau s=s+c -> prelucrare cifra

n/=10; // sau n=n/10

}

cout<<s;

return 0;

}

**P2) Prima cifra a lui n**

cin>>n;

while (n>9) n/=10;

cout<<n;

**P3) Numarul de cifre pare ale lui n**

cin>>n;

do {

if (n%2==0) nr++; // c=n%10; if (c%2==0) nr++;

n/=10;

} while(n); // while (n>0);

cout<<nr;

**P4) Sa se verifice daca un numar natural n este palindrom ( este egal cu oglinditul sau)**

**Exemplu:** n=2431 -> c=1, on=1

c=3, on=13 -> on=on\*10+c

c=4, on=134

c=2, on=1342

cin >>n;

on=0;

aux=n; // aux= copia lui n

while (aux)

{ c=aux%10;

on=on\*10+c;

aux/=10;

}

if (n==on) cout<<”DA”;

else cout<<”NU”;

**P5) Sa se stearga din numarul n toate cifrele impare -> Construirea unui nou numar x ccu cifrele pare ale lui n, in ordine de la stanga la dreapta.**

**Ex.**

n=23456873 -> x=2468

c=3

c=7

c=8 -> x=8, p=10

c=6 -> x=68, p=102 -> x=c\*p + x, p=p\*10;

c=5

c=4 -> x=468, p=103

c=3

c=2 -> x=2468, p=104

cin >>n;

x=0; p=1; //100

while (n>0)

{ c=n%10;

if (c%2==0)

{ x=c\*p + x; p=p\*10;}

n/=10;

}

cout<<x

**Tema: 3077, 3078, 66, 2288, 3254, 69, 77, 108, 118, 121, 871, 1613, 3079, 1582, 127**

**3254)**

#include<iostream>

using namespace std;

int n, mn=9, mx=0, c;

int main()

{ cin>>n;

while(n>0)

{c=n%10;

if (c%2!=0 && c<mn) mn=c;

if (c%2==0 && c>mx) mx=c;

n/=10;

}

cout<<mn\*mx;

return 0;

}

**3078)**

#include <iostream>

using namespace std;

int main()

{

int n, p = 1, k, c, ok=0;

cin >> n >> k;

do{

c = n % 10;

if(c != k)

{ p = p \* c;

ok = 1;

}

n = n / 10;

}while(n);

if(!ok) // if (ok==0)

cout << 0;

else cout << p;

return 0;

}

**3077)**

#include <iostream>

using namespace std;

**long long** n, s;

int main()

{

cin >> n;

while (n>0)

{

s = s + n;

n = n / 10;

}

cout << s;

return 0;

}

**871)**

#include <iostream>

using namespace std;

int n , c1, c2, c, x, p=1;

int main()

{

cin >> n >> c1 >> c2;

do{ c=n%10;

if (c==c1) c=c2;

x=c\*p+x;

p\*=10;

n/=10;

} while(n);

cout << x;

return 0;

}

**1613)**

#include <iostream>

using namespace std;

int n,n1,n2,c,p1=1,p2=1;

int main ()

{

cin>>n;

while(n)

{

c=n%10;

if(c%2==0) {n1=n1+c\*p1; p1=p1\*10;}

else {n2=n2+c\*p2; p2=p2\*10;}

n=n/10;

}

if(n1>n2) cout<<n1-n2;

else cout<<n2-n1;

return 0;

}

1. **Numararea divizorilor pozitivi ai lui n.**

**Obs. Toti divizorii sunt intre 1 si n**

**Divizorii propri sunt intre 2 si n/2**

**Solutia 1 –> O(n)**

cin>>n;

nr=0;

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

if (n%d==0) nr++;

cout<<nr;

**Solutia 2 –> O() – eficienta ca timp de executare**

Exemplu:

n=100

d n/d

1 100

2 50

4 25

5 20

10

cin>>n;

nr=0;

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

if (n%d==0) **nr+=2**;

**if (d\*d==n) nr++; daca n e patrat perfect**

cout<<nr;

**P1) Suma divizorilor pozitivi ai lui n.**

cin>>n;

s=0;

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

if (n%d==0) **s=s+d+n/d**;

**if (d\*d==n) s+=d; daca n e patrat perfect**

cout<<s;

1. **Sa se verifice daca n este numar prim**

cin>>n;

**ok=1; // pp. ca n este numar prim**

if(n<2) ok=0;

for (d=2; d\*d<=n; d++) // sau d<=n/2

if (n%d==0)

{ok=0;

break;

}

if(ok) cout<<”DA”;

else cout<<”NU”;

**Tema:**

**divizori: 376, 387, 388, 1568, 389, 61, 408, 64**

**nr.prime: 45, 373, 379, 381, 404, 407, 375, 116**

**61)**

**Sol 1**

using namespace std;

int a,b,d,aux;

int main()

{

cin>>a>>b; // mn=a; if(b<a) mn=b;

if(a>b) {

aux=a;

a=b;

b=aux;

}

for(d=1; d<=a; d++) // for(d=1; d<=mn; d++)

if(a%d==0 && b%d==0) cout<<d<<' ';

return 0;

}

**Sol 2 – eficienta ca timp**

using namespace std;

int a,b,d,r;

int main()

{

cin>>a>>b;

while(b>0) // cmmmdc(a.b)

{

r=a%b;

a=b;

b=r;

}

for(d=1; d<=a; d++)

if(a%d==0) cout<<d<<' ';

return 0;

}

**389)**

using namespace std;

int i,d,a,b,nr,mx,MIN,MAX;

int main()

{

cin>>a>>b;

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

{

nr=0; // nr= numarul de divizori pari ai lui i

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

if(i%d==0)

{

if(d%2==0) nr++;

if((i/d)%2==0) nr++;

}

if(d\*d==i && d%2==0) nr++;

if(nr>mx) {MIN=MAX=i; mx=nr;}

else if(nr==mx) MAX=i;

}

cout<<mx<<' '<<MIN<<' '<<MAX;

return 0;

}

**116)**

#include <iostream>

#include <iomanip>

#include <cmath>

using namespace std;

int n,d,ok,x,nr,i;

float s, ma;

int main()

{

cin>>n;

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

{

cin>>x;

ok=1; // verific daca x este numar prim

if (x<2) ok=0;

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

if(x%d==0) {ok=0; break;}

if(ok) {s=s+x; nr++;}

}

**// pastrarea a 2 zecimale exacte**

ma=s/nr;

ma=floor(ma\*100)/100;

**// afisare cu 2 zecimale**

cout<< setprecision(2) << fixed << ma;

return 0;

}

**381)**

int n,x,i,k,d,nr,M,d1,ok,k1;

int main()

{

cin>>n;

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

{

cin>>x;

// cel mai mic divizor prim al lui x este cel mai mic divizor al lui x diferit de 1

k1=0;

for(k=2; k\*k<x; k++)

if (x%k == 0)

{ k1=k; break;}

if(k1 && x/k1!=k1)

{d1=x/k1; // x=k\*d1, cu k si d1 numere prime si diferite

ok=1; // verific daca d1 e numar prim

if (d1<2) ok=0;

for(d=2; d\*d<=d1; d++)

if(d1%d==0) {ok=0; break;}

if(ok)

if(x>M) {M=x; nr=1;}

else if(x==M) nr++;

}

}

cout<<M<<' '<<nr;

return 0;

}

**375)**

int n,i,ok,d;

int main()

{

cin>>n;

i=n+1;

do

{ ok=1;

for(d=2;d\*d<=i;d++)

if(i%d==0) {ok=0; break;}

if (!ok) i++;

} while(!ok);

cout<<i;

return 0;

}

1. **cmmdc(a,b) si cmmmc(a,b)**
2. **prin scaderi succesive**

cin>>a>>b;

while(a!=b)

if (a>b) a-=b;

else b-=a;

cout<<a;

1. **algoritmul lui Euclid**

**a=30, b=26**

**a b r**

**30 = 26 \* 1 + 4**

**26 = 4 \* 6 + 2**

**4 = 2 \* 2 + 0**

**a=2, b=0**

cin>>a>>b;

p=a\*b;

while (b) // while(b>0)

{ r=a%b;

a=b;

b=r;

}

cout<<a<<endl; // cmmdc(a,b)

cout<<p/a<<endl; // cmmmc(a,b)

**Obs. cmmmc(a,b)= a\*b/cmmdc(a,b)**

**Tema: 58, 59, 3268, 60, 409, 305, 390, 391, 80**

**305)**

int n,a,b,r,i;

int main()

{

cin>>n;

cin>>a;

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

{

cin>>b;

while(b)

{

r=a%b;

a=b;

b=r;

}

}

cout<<a;

return 0;

}

**390)**

**Obs. a/b+c/d = (a\*d+c\*b) / b\*d = 30/40**

int a,b,c,d,s1,s2,r,x,y;

int main()

{

cin>>a>>b>>c>>d;

s1=a\*d+c\*b; // numaratorul fractiei suma

s2=b\*d; // numitorul fractiei suma

x=s1;

y=s2;

while(y)

{

r=x%y;

x=y;

y=r;

}

s1=s1/x;

s2=s2/x;

cout<<s1<<' '<<s2<<endl;

a=a\*c; // numaratorul fractiei produs

b=b\*d; // numitorul fractiei produs

x=a;

y=b;

while(y)

{

r=x%y;

x=y;

y=r;

}

a=a/x;

b=b/x;

cout<<a<<' '<<b;

return 0;

}

**391)**

int n,a,b,c,d,r,i,p,q;

int main()

{

cin>>n;

cin>>a>>b;

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

{

cin>>c>>d; //a/b+c/d-> a/b

a=c\*b+a\*d;

b=d\*b;

p=a;

q=b;

while(q)

{

r=p%q;

p=q;

q=r;

}

a=a/p; b=b/p;

}

cout<<a<<" "<<b;

return 0;

}

1. **Sirul lui Fibonacci**

**f(1)=f(2)=1;**

**f(n)=f(n-1)+f(n-2)**

**1 1 2 3 5 8 13 21 34**

**Sa se determine termenul de rang n al sirului.**

cin>>n;

f1=f2=1;

if (n==1 || n==2) cout<<1;

else { for(i=3; i<=n; i++)

{ f3=f1+f2;

f1=f2;

f2=f3;

}

cout<<f3;

}

**P1) Sa se verifice daca numarul x este termen in sirul lui Fibonacci**

Ex. x=13 -> DA

x=12 -> NU

cin>>x;

f1=f2=1;

while (f2<x)

{ f3=f1+f2;

f1=f2;

f2=f3;

}

if (f2==x) cout<<”DA”;

else cout<<”NU”;

**Tema: 255, 423, 256, 257, 2793, 806**

**423)**

int n,f1,f2,f3;

int main()

{

cin>>n;

f1=f2=1;

while(f1<=n)

{

cout<<f1<<' ';

f3=f1+f2;

f1=f2;

f2=f3;

}

return 0;

}

**257)**

**Ex.**

**1 1 2 3 5 8 13 21 34**

**n=20**

**x=13, scrie 13, n=20-13=7**

**x=5, scrie 5, n=7-5=2**

**x=2, scrie 2, n=2-2=0**

int main()

{

int f1, f2, n, f3;

cin>>n;

while (n>0)

{ // determin cel mai mare termen din sir <=n, in f1

f1=1;

f2=1;

while (f2<=n)

{

f3=f1+f2;

f1=f2;

f2=f3;

}

cout<<f1<<" ";

n=n-f1;

}

return 0;

}

**2793)**

**Y=f(n)**

**X=f(n-1)**

**Z=f(n-2)**

**(y, x, z) ultimii trei termeni -> Y=2\*x-z+2 -> z=-y+2\*x+2**

cin>>x>>y;

f<<y<<” ”<<x<<” ”;

while(x>0)

{ z=-y+2\*x+2;

f<<z<<” ”;

y=x; x=z;

}

**Sol II**

**F(n)=n2-1 -> n2-1=y -> n=radical(y+1)**

n=sqrt(y+1);

for (i=n; i>=1; i--)

g<<i\*i-1<<” ”;

**806)**

int x,p,q;

int main()

{

cin>>x;

while(**x>1**)

{

cout<<x<<" ";

if(x%2!=0) x=(x+1)/2;

else x=x-1;

}

**cout<<1;**

return 0;

}

1. **Descompunerea unui numar in factori primi**

N=700

n f

700 2 -> p=1

350 2 -> p=2

175 5 -> p=1

35 5 -> p=2

7 7 -> p=1

1

**P1) Afisarea factorilor primi si a puterilor la care apar in descompunere (O(n))**

cin>>n;

f=2;

while (n>1)

{ p=0;

while (n%f==0)

{ n=n/f;

p++;

}

if (p) cout<<f<<” ”<<p<<’\n’;

f++;

}

**P2) numarul de factori primi din descompunere (O(n))**

cin>>n;

f=2;

while (n>1)

{ if ( n% f==0)

{ nr++;

while (n%f==0) n=n/f;

}

f++;

}

**Obs. Algoritmul eficient ca timp (O)**

**P1) Afisarea factorilor primi si a puterilor la care apar in descompunere**

cin>>n;

f=2;

while (n>1 && f\*f<=n)

{ p=0;

while (n%f==0)

{ n=n/f;

p++;

}

if (p) cout<<f<<” ”<<p<<’\n’;

f++;

}

if(n>1) cout<<n<<” ”<<1<<’\n’;

**Ex. n=66=2\*3\*11**

**Tema: 1319, 62, 3352, 63, 111, 2821, 435, 463**

**1319)**

int main()

{ int n,f;

cin>>n;

f=2;

while (n>1 && f\*f<=n)

{

while (n%f==0)

{n=n/f;

cout<<f<<" ";

}

f++;

}

if(n>1) cout<<n;

return 0;

}

**Siruri de numere citite de la tastatura**

**1) Prelucrarea unui sir de n numere citite de la tastatura - sir cu numar cunoscut de elemente.**

cin>>n;

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

{ cin>>x;

// prelucreaza x;

}

**2) Prelucrarea unui sir de numere cu numar necunoscut de elemente. Ex. Se citeste un sir de numere pana la intalnirea valorii 0 care nu face parte din sir. Se cere prelucrarea numerelor din sir.**

**Exemplu: 20, 67, 44, 89, 90, 0**

cin>>x;

while (x!=0)

{ // prelucreaza x;

cin>>x;

}

**3) Prelucrarea perechilor de numere citite consecutiv, dintr-un sir de n numere citite de la tastatura.**

cin>>n>>x;

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

{ cin>>y;

// prelucreaza perechea (x,y)

x=y; // penultimul devine ultimul citit -> trecerea pe urmatoarea pereche

}

**Tema1: 56, 277, 347, 54, 55, 171, 346, 2269,** **119, 274, 3284, 172, 2275, 282**

**409, 305, 381, 404, 407, 116, 1582, 127**

**3352)**

int a,f,s1,b,s2,x,y;

int main()

{

cin>>a>>b;

x=a; y=b;

f=2;

while(a>1 && f\*f<=a)

{

if (a%f==0)

{ s1+=f;

while(a%f==0) a=a/f;

}

f++;

}

if(a>1) s1=s1+a;

f=2;

while(b>1 && f\*f<=b)

{

if (b%f==0)

{ s2+=f;

while(b%f==0) b=b/f;

}

f++;

}

if(b>1) s2=s2+b;

if(s1>s2) cout<<x;

else if(s1<s2) cout<<y;

else if(x<y) cout<<x;

else cout<<y;

return 0;

}

**63)**

int d,x,p,mp,y;

int main()

{

cin>>x;

d=2;

while(x>1 && d\*d<=x)

{

p=0;

while(x%d==0) {x=x/d; p++;}

if(p>=mp) {mp=p; y=d;}

d++;

}

if(x>1 && mp<=1) y=x;

cout<<y;

return 0;

}

**111)**

int n, f = 2, mx , r, x, p, i, cx;

int main()

{

cin >> n;

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

{

cin >> x;

cx = x;

p=0; // p=numarul de factori primi ai lui x

f=2;

while(x > 1 && f \* f <= x)

{

if (x % f == 0)

{ p++;

while(x % f == 0) x = x / f;

}

f++;

}

if(x > 1) p++;

if (p > mx) { mx = p; r = cx;}

else if (p == mx && cx < r) r = cx;

}

cout << r;

return 0;

}

**463)**

**2\*32  2\*5 2\*3 3\*54**

**Fmx 3 5**

**Pmx 2 1 5**

int n , x, mx, exp, p, d;

int main()

{

cin >> n;

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

{

cin >> x;

d=2;

while(x > 1 && d \* d <= x)

{

p = 0;

while(x % d== 0)

{ p++; x /= d; }

if(p)

if(d > mx) { mx = d; exp = p;}

else if(d == mx) exp += p;

d ++;

}

if(x > 1)

if(x > mx) { mx = x; exp = 1;}

else if(x == mx) exp++;

}

cout << mx << " " << exp;

return 0;

}

**171)**

int main()

{

int n, m = 9, x1, maxx = 0, x;

cin >> n;

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

{

cin >> x;

x1 = x;

while(x > 9)

x = x / 10;

if(x < m) { m = x; maxx = x1;}

else if (x == m && x1>maxx) maxx=x1;

}

cout << maxx;

return 0;

}

**56)**

int a, b, nr;

int main()

{

cin>>a; a=abs(a);

if (a)

{cin>>b; b=abs(b);

while(b!=0)

{

if(b%10==a%10) nr++;

a=b;

cin>>b; b=abs(b);

}

}

cout<<nr;

return 0;

}

**3284)**

int n, mn, mx, x, i;

int main()

{

cin>>n;

mn=100;

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

{

cin>>x;

if(i%2==0 && x>mx)

mx=x;

if(i%2!=0 && x>9 && x<100 && x<mn)

mn=x;

}

if(mn==100) mn=-1;

cout<<mx<<" "<<mn;

return 0;

}

**2275)**

ifstream fin("minimdoua.in");

ofstream fout("minimdoua.out");

int main()

{

int n, x, min1 = 2000000, min2 = 2000000;

fin >> n;

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

{

fin >> x;

if (x > 9 && x%10 == x/10%10)

if (x < min1) { min2 = min1; min1 = x;}

else if (x < min2) min2=x;

}

if (min2 == 2000000 || min1 == 2000000)

fout << "numere insuficiente";

else fout << min1 << ' ' << min2;

return 0;

}

**Tema2: 132, 248, 249, 289, 506, 288, 501, 500, 290, 291, 293, 294**

**132)**

ifstream f("elempp.in");

ofstream g("elempp.out");

int n,x,i,y;

int main()

{

f>>n;

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

{

f>>x;

y=x;

while(y>99) y/=10;

if(sqrt(y)==floor(sqrt(y)))g<<x<<" ";

}

f.close();

g.close();

return 0;

}

**248)**

ifstream fin("pozitie.in");

ofstream fout("pozitie.out");

int n , x , a , p ;

int main()

{

fin >> n >> x;

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

{

fin >> a;

if(a < x) p++;

}

fout << p + 1;

return 0;

}

**249)**

ifstream f("pozitiex.in");

ofstream g("pozitiex.out");

int n,i,x,nr,b,OK;

int main()

{

f>>x>>n;

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

{

f>>b;

if(b==x) OK=1;

else if(b<x) nr++;

}

if(OK) g<<nr + 1;

else g<<"NU EXISTA";

return 0;

}

**506)**

int n,a,ok,i,p,j;

int main()

{

cin>>n;

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

{

cin>>a;

ok=1;

if (a<2) ok=0;

for(j=2;j\*j<=a;j++)

if(a%j==0)

{

ok=0;

break;

}

if(ok)

p++;

}

if (p) cout<<"DA";

else cout<<"NU";

return 0;

}

**290)**

int n,i,x=-10000,ok=1,y;

int main()

{

cin>>n;

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

{

cin>>y; // x= penultimul par citit; y=ultimul par citit

if(y%2==0)

if(y<x) {ok=0; break;}

else x=y;

}

if(ok) cout<<"DA";

else cout<<"NU";

return 0;

}

**293)**

**Sol 1**

int n,ok,a,x,u,i;

int main()

{

cin>>n;

ok=1;

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

{

cin>>a;

while(a!=0)

{ u=a%10; x=a/10;

while(x!=0)

{

if(u==x%10) ok=0;

x=x/10;

}

a=a/10;

}

}

if(ok)cout<<"DA";

else cout<<"NU";

return 0;

}

**Sol 2 – cu vector frecvente cifre**

int main()

{

int n, x, cif[10]; // cif[c]= numarul de aparitii al cifrei c in numarul x

cin >> n;

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

{

cin >> x;

for(int i = 0; i < 10; i++) cif[i] = 0;

while(x > 0)

{

c = x % 10;

cif[c]++;

if(cif[c] > 1)

{

cout << "NU";

return 0;

}

x = x / 10;

}

}

cout << "DA";

return 0;

}

**294)**

int n,i,x,inv,ok,c;

int main()

{

cin>>n>>x;

while(x!=0)

{

c=x%10;

inv=inv\*10+c;

x=x/10;

}

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

{ cin>>x;

if(x==inv)

{ok=1; break;}

}

if(ok)cout<<"DA";

else cout<<"NU";

return 0;

}

**Secvente de lungime maxima cu o proprietate data**

1. Se da un sir de n numere naturale. Sa se determine cea mai lunga secventa de numere **pare aflate pe pozitii consecutive in sir. Se va afisa lungimea maxima a secventei.**

**Ex. n=10**

**2 4 3 7 8 10 12 14 15 16**

**l 1 2 0 0 1 2 3 4 0 1**

int n, x, l, lmax; // var globale

cin>>n;

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

{ cin>>x;

if (x%2==0)

{ l++;

if (l>lmax) lmax=l;

}

else l=0; // lungimea secventei curente se reintializeaza cu 0

}

cout<<lmax;

**Obs. Algoritm eficient ca timp de executare, fiind un algoritm liniar de complexitate O(n).**

1. **Problema platoului. Se da un sir de n numere naturale. Sa se determine cea mai lunga secventa de numere egale aflate pe pozitii consecutive in sir. Se va afisa lungimea maxima a secventei si numarul din care este formata cea mai din stanga secventa de lungime maxima.**

**Ex. n=13**

**1 1 2 2 2 2 3 3 3 5 5 5 5**

**l 1 2 1 2 3 4 1 2 3 1 2 3 4**

**Afiseaza: 4 2**

int n, x, l, lmax; // var globale

cin>>n;

**cin>>x; // se citeste primul numar din sir**

**l=lmax=1; // 1 2 3**

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

{ cin>>y;

if (x==y)

{ l++;

if (l>lmax) { lmax=l; r=y; }

}

**else l=1; // lungimea secventei curente se reintializeaza cu 1**

**x=y; // trecerea la urmatoarea pereche**

}

cout<<lmax<<” ”<<r;

1. **Obs. Se va afisa lungimea maxima a secventei si numarul din care este formata cea mai din dreapta secventa de lungime maxima.**

if (**l>=lmax**) { lmax=l; r=y; }

**Tema: 1875, 1879, 1880, 1881**

**1875)**

ifstream f("platou1.in");

ofstream g("platou1.out");

int main()

{

int xmin,x,y,l=1,mx=1;

f>>x; xmin=x;

while (f>>y)

{

if (x==y) l++;

else l=1;

if (l>mx) {mx=l;xmin=y;}

else if (l==mx && y<xmin) xmin=y;

x=y;

}

g<<mx<<" "<<xmin;

f.close();

g.close();

return 0;

}

**1879)**

**Sol I)**

fstream f("platou2.in");

ofstream g("platou2.out");

int main()

{

int n,i,x,y,mx=1,l=1;

f>>n>>x;

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

{

f>>y;

if(x<y)

{

l++;

if(l>mx) mx=l;

}

else l=1;

x=y;

}

g<<mx;

return 0;

}

**Sol II)**

fstream f("platou2.in");

ofstream g("platou2.out");

int main() // 2 3 1 5 6 4 6 7 8

{

int n,i,x,y,mx=1,l=1;

f>>n>>x;

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

{

f>>y;

if(x<y) l++;

else {if(l>mx) mx=l; l=1;}

x=y;

}

if(l>mx) mx=l; // prelucrarea ultimei secvente de numere strict crescatoare

g<<mx;

return 0;

}

**1880)**

fstream f("platou3.in");

ofstream g("platou3.out");

int main()

{

int n,i,x,mx=0,l=0;

f>>n;

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

{

f>>x;

if(x>0)

{

l++;

if(l>mx) mx=l;

}

else l=0;

}

g<<mx;

return 0;

}

**1881)**

ifstream f("platou4.in");

ofstream g("platou4.out");

int x, k, p, u;

int main()

{

while(f >> x)

{

k++;

if (x%2==0)

{ if (p==0) p=k;

u=k;

}

}

g << u-p+1;

return 0;

}