



PROBLEME

STRUCTURI REPETITIVE

Scrie un program care, citind un număr natural  $n$ , afișează suma cifrelor acestuia. **Exemplu:** pentru  $n = 283$  se afișează 13.

while	for	do while
<pre>int n,s=0; cin&gt;&gt;n; while(n&gt;0) {     s=s+n%10;     n=n/10; } cout&lt;&lt;s;</pre>	<pre>int n,s,i; cin&gt;&gt;n; for(s=0;n&gt;0;n=n/10)     s=s+n%10; cout&lt;&lt;s; return 0;</pre>	<pre>int n,s=0,i; cin&gt;&gt;n; do {     s=s+n%10;     n=n/10; } while(n&gt;0); cout&lt;&lt;s;</pre>

## EXPLICATII

```
int n,s=0;
cin>>n;
while(n>0)
{
    s=s+n%10;
    n=n/10;
}
cout<<s;
```


Inițializez variabila **s** cu 0

Citesc variabila **n**

Cat timp **n** este mai mare decat 0, prelucrez cifrele  
numarului **n**:

Adaug la suma restul impartirii numarului **n** la 10 ( $n\%10$ ) și  
elimin ultima cifra a numarului ( $n=n/10$ ) (în final **n** va ajunge  
0 – ceea ce este condiție de ieșire din **while**)

Afișez suma **s**.



Modifică programele de mai sus, astfel încât să calculeze suma cifrelor pare ale numărului citit.

**Exemplu:** pentru  $n=2359681$  se afișează 16.



Determină ce se afișează în urma rulării celor două secvențe de program

secvență	valori afisate
<pre>int a=259; cout&lt;&lt;a/10&lt;&lt;" "&lt;&lt;a%10;</pre>	
<pre>int b=3206; cout&lt;&lt;b/10&lt;&lt;" "&lt;&lt;b%10;</pre>	

- Câtul împărțirii la 10 este numărul fără ultima cifră
- Restul împărțirii la 10 este ultima cifră a numărului

Scrieți un program care numără câte cifre impare sunt într-un număr natural  $n$ . Spre exemplu, pentru  $n=2359687$  se afișează 4

while	for	do while
<pre>int n, nr=0; cin&gt;&gt;n; while(n&gt;0) {     if(n%10%2==1)         nr++;     n=n/10; } cout&lt;&lt;nr;</pre>	<pre>int n, nr=0; cin&gt;&gt;n; for(nr=0;n&gt;0;n=n/10) {     if(n%10%2==1)         nr++; } cout&lt;&lt;nr;</pre>	<pre>int n, nr=0; cin&gt;&gt;n; do {     if(n%10%2==1)         nr++;     n=n/10; } while(n&gt;0); cout&lt;&lt;nr;</pre>

Scrie un program care, citind un număr natural  $n$ , afișează câte cifre are. **Exemplu:** pentru  $n = 275$  se afișează 3.

while	for	do while
<pre>int n,nr=0; cin&gt;&gt;n; while(n&gt;0) {     n=n/10;     nr++; } cout&lt;&lt;nr;</pre>	<pre>int n,nr; cin&gt;&gt;n; for(nr=0; n&gt;0; n=n/10)     nr++; cout&lt;&lt;nr;</pre>	<pre>int n,nr=0; cin&gt;&gt;n; do {     n=n/10;     nr++; } while(n&gt;0); cout&lt;&lt;nr;</pre>



○ Descrue un program care determină cea mai mare cifră a unui număr natural **n**.

○ Scrieți programul folosind și celelalte două tipuri de structuri repetitive din C++ (for și do-while).

```
int n,maxx;
cin>>n;
maxx=n%10;
while(n>0)
{
    if(n%10>maxx)
        maxx=n%10;
    n=n/10;
}
cout<<maxx;
```

Describe un algoritm care afișează divizorii comuni a două numere naturale  $n$  și  $m$ .

**Exemplu:** pentru  $n = 180$  și  $m = 105$  se afișează 3 5 15.

```
int n,m,d;
cin>>n>>m;
if(n<m)
    swap(n,m);
for(d=2;d<=n;d++)
    if(n%d==0 && m%d==0)
        cout<<d<<" ";
```

citim  $n$  și  $m$

Dacă  $n < m$ , schimbăm între ele valorile celor două variabile

Cea mai mare valoare este  $n$ . Căutăm prin divizorii lui  $n$ , valori ( $d$ ) care să fie divizor și pentru  $m$ .

Dacă  $d$  este divizor comun, îl afișăm

Scrive un program care, citind un număr natural  $n$ , creează un număr  $m$  ce conține cifrele numărului  $n$  în ordine inversă.

**Ex.:** Pentru  $n = 123$  se obține  $m = 321$ .

**Indicație:**

```
int n,m=0,c;  
cin>>n;  
while(n>0)  
{  
    c=n%10;  
    m=m*10+c;  
    n=n/10;  
}  
cout<<m;
```

Păstrez în variabila  $c$  ultima cifră a numărului, calculată cu  $n\%10$  și pentru a adăuga o cifră  $c$  la sfârșitul unui număr  $m$ , scrii  $m = m*10 + c$ .  
Elimin ultima cifră a numărului

Andrei consideră că un număr e cu atât mai bogat cu cât are mai mulți divizori. Scrie un program care să-i indice lui Andrei care dintre cele două numere naturale citite este cel mai bogat.

**Exemplu:** dacă cele două numere sunt 6 și 7 se afișează 6.

```
int n,m,nrn=0,nrm=0,d,e;
cin>>n>>m;
for(d=2; d<=n/2; d++)
    if(n%d==0)
        nrn++;
for(d=2; d<=m/2; d++)
    if(m%d==0)
        nrm++;
if(nrn>nrm)
    cout<<n;
else
    cout<<m;
```

**nrn** = numărul de divizori ai lui **n**  
**nrm** = numărul de divizori ai lui **m**  
căutam divizorii lui **n**  
și creștem **nrn** dacă găsim unul

căutam divizorii lui **m**  
și creștem **nrm** dacă găsim unul

comparam **nrn** cu **nrm**

Scrive un program care determină dacă un număr natural  $n$  este prim sau nu.

```
int n,prim,d;  
cin>>n;  
prim=1;  
for(d=2;d<=n/2;d++)  
    if(n%d==0)  
        prim=0;  
if(prim)  
    cout<<"prim";  
else  
    cout<<"nu e prim";
```

Consider că numărul  $n$  citit este prim ( $\text{prim}=1$ )

Caut divizori ( $d$ ) începând cu 2 până la  $n/2$   
Dacă  $n$  se divide cu  $d$  atunci numărul nu este prim  
( $\text{prim}=0$ )

Afisez în funcție de valoarea lui  $\text{prim}$  (1 sau 0) că  
numărul este prim (1) sau nu (0)

# Scrive un program care afișează descompunerea în factori primi a unui număr natural $n$ .

```
#include <iostream>
using namespace std;
int main()
{
    int n, d=2, p=0;
    cout<<"n:"; cin>>n;
    while(n!=1)
    {
        p=0;
        while(n%d==0)
        {
            p++;
            n=n/d;
        }
        if(p!=0)
            cout<<d <<" la puterea " <<p<<endl;
        d++;
    }
    return 0;
}
```

Se citește numărul  $n$ , iar apoi se pornește cu fiecare număr de la 2 până când variabila introdusă ajunge să fie 1.

Cât timp  $n\%d==0$  se împarte numărul la  $d$ , iar puterea ( $p$ ) crește cu o unitate.

La final, se afișează divizorul și puterea acestuia.

Pentru secvența de program de mai jos, răspunde următoarelor cerințe:

- a) determină ce afișează pentru  $n = 9$ ;
- b) determină ce afișează pentru  $n = 17$ ;
- c) rescrie secvența folosind **for**;
- d) scrie un program echivalent (care afișează exact aceleași valori ca cel din exemplu, pentru aceleași date de intrare) care să nu folosească structuri repetitive.

```
int n, i;  
cout <<"Dati n:";  
cin >> n;  
i=1;  
while (i*i<=n) i=i+1;  
cout << i-1;  
return 0;
```

n	i	i*i	i-1
9	1	1	
	2	4	
	3	9	
	4	16	3

```
int n,i;  
cin>>n;  
for(i=1;i*i<=n;i++);  
cout<<i-1<<endl;
```

Scrive un program care, citind două numere naturale **a** și **b**, afișează cel mai mare divizor comun al lor.

```
int a,b;  
cin>>a>>b;  
do  
{  
    if(a>b)  
        a=a-b;  
    else  
        b=b-a;  
}  
while(a!=b);  
cout<<a;
```

se determina cel mai mic divizor comun prin scaderi repetate

se afiseaza a (care memoreaza acum cmmdc-ul dintre cele 2 numere)