

Structura repetitivă cu număr cunoscut de pași. Divizibilitate. Primaritate

Structura repetitivă cu număr cunoscut de pași. Divizibilitate. Primaritate

- ▶ **Structura repetitivă cu număr cunoscut de pași**
 - Structuri repetitive
 - Structura repetitivă cu contor (cu număr cunoscut de pași)
 - Instrucțiunea *for*
 - ▶ **Divizibilitate**
 - Noțiunea de divizor
 - Implementare în limbajul C++
 - ▶ **Primaritate**
 - Noțiunea de număr prim
 - Implementare în limbajul C++
 - ▶ **Probleme propuse**
 - ▶ **Bibliografie**
-
- ▶ **Structura repetitivă cu număr cunoscut de pași**

Structuri repetitive

Repetiția (bucla, ciclul sau iterația) asigură execuția unei secvențe în mod repetat în funcție de o anumită condiție.

Există trei tipuri de structuri repetitive:

- structura repetitivă *cu test inițial*;
- structura repetitivă *cu test final*;
- structura repetitivă *cu contor*.

Structura repetitivă cu contor (cu număr cunoscut de pași)

- execuția se descrie astfel:

```
pentru contor=val_inițială, val_finală, pas execută  
    secvența A
```

Principiul de execuție:

Pasul 1: variabila de ciclare `contor` primește `val_inițială`

Pasul 2: dacă `contor` este mai mic sau egal cu valoarea finală `val_finală`, se execută Secvența A, se adună valoarea pasului la `contor` și se reia cu pasul 2, altfel, execuția este încheiată.

Observație: dacă valoarea pasului este 1, nu este obligatorie specificarea pasului.

Structura repetitivă cu număr cunoscut de pași. Divizibilitate. Primaritate

Exemplu:

Să se calculeze și să se afișeze pe ecran suma primelor n numere naturale nenule. Valoarea lui n (număr natural nenul) se citește de la tastatură.

Date de intrare: n

Date de ieșire: s

Date de manevră: i

```
citește n (număr natural nenul)
S ← 0
pentru i=1, n execută
  ■ S ← S+i
scrie S
```

Instrucțiunea for

În limbajul de programare C++ structura repetitivă cu contor este implementată cu ajutorul instrucțiunii *for*. Instrucțiunea *for* este o instrucțiune repetitivă cu număr cunoscut de pași.

Sintaxa:

```
for (expresie1; expresie2; expresie3)
    instrucțiune;
```

Principiul de execuție:

Pasul 1: se evaluează expresie₁;

Pasul 2: se evaluează expresie₂; dacă aceasta produce o valoare diferită de 0, se execută instrucțiune, apoi se trece la pasul 3, altfel instrucțiunea *for* se încheie;

Pasul 3: se evaluează expresie₃ și se revine la pasul 2.

Exemplu:

Să se calculeze și să se afișeze pe ecran suma primelor n numere naturale nenule. Valoarea lui n (număr natural nenul) se citește de la tastatură.

Date de intrare: n

Date de ieșire: s

Date de manevră: i

```
#include <iostream>
```

```
using namespace std;
```

Structura repetitivă cu număr cunoscut de pași. Divizibilitate. Primaritate

```
int main()
{
    int n,s,i;
    cin>>n;
    s=0;
    for(i=1;i<=n;i++)
        s=s+i;
    cout<<s;
    return 0;
}
```

► Divizibilitate

Noțiunea de divizor

Fie n număr natural nenul. Dacă d divide pe n , atunci d este divizor al lui n .

Divizorii unui număr natural n sunt acele numere naturale nenule d pentru care se obține restul 0 la împărțirea lui n la d :

$$n \% d == 0$$

unde d este număr natural nenul.

Exemple:

1. $n = 18$

- divizorii lui 18: 1, 2, 3, 6, 9 și 18
- divizorii improprii ai lui 18: 1 și 18
- divizorii proprii ai lui 18: 2, 3, 6 și 9

2. $n = 23$

- divizorii lui 23: 1 și 23
- divizorii improprii ai lui 23: 1 și 23
- divizorii proprii ai lui 23: nu există

3. $n = 25$

- divizorii lui 25: 1, 5 și 25
- divizorii improprii ai lui 25: 1 și 25
- divizorii proprii ai lui 25: 5

Implementarea în limbajul C++

Să se afișeze pe ecran divizorii unui număr natural nenul n , citit de la tastatură.

Varianta 1

```
cin>>n;
for(d=1;d<=n;d++)
    if(n%d==0)
        cout<<d<<' ';
```

Structura repetitivă cu număr cunoscut de pași. Divizibilitate. Primaritate

Varianta 2

```
cin>>n;
for (d=1; d<=n/2; d++)
    if (n%d==0)
        cout<<d<<' ';
cout<<n;
```

Varianta 3

```
cin>>n;
for (d=1; d*d<=n; d++)
{
    if (n%d==0)
    {
        cout<<d<<' ';
        if (d!=n/d)
            cout<<n/d<<' ';
    }
}
```

► Primaritate

Noțiunea de număr prim

Un număr n este prim dacă nu are alți divizori în afară de 1 și n , (nu are divizori proprii).

Exemple:

1. $n = 11$

- divizorii lui 11: 1 și 11
- numărul 11 este prim

2. $n = 15$

- divizorii lui 15: 1, 5 și 15
- numărul 15 nu este prim

Implementarea în limbajul C++

Să se afișeze pe ecran divizorii unui număr natural nenul n , citit de la tastatură.

Varianta 1

```
cin>>n;
nr_div=0;
for (d=2; d<=n/2; d++)
    if (n%d==0)
        nr_div++;
if (nr_div==0 && n>=2)
    cout<<"numarul este prim";
else
    cout<<"numarul nu este prim";
```

Structura repetitivă cu număr cunoscut de pași. Divizibilitate. Primaritate

Varianta 2

```
cin>>n;  
prim=1;  
if(n<=1)  
    prim=0;  
for(d=2;d*d<=n;d++)  
    if(n%d==0)  
        prim=0;  
if(prim==1)  
    cout<<"numarul este prim";  
else  
    cout<<"numarul nu este prim";
```

► Probleme propuse

Pbinfo:

☒ Probleme:

#376, #387, #388, #374, #373

► Bibliografie

1. Cerchez E., Șerban M., *Programarea în limbajul C/C++ pentru liceu*, Editura Polirom, Iași, 2005
2. <https://www.pbinfo.ro/>
3. <https://www.ududec.com/>