

Structura repetitivă cu număr cunoscut de pași

Forma instrucțiunii **for** este:

for (Expresie_de_Initializare; Expresie_de_Testare; Expresie_de_Continuare)

Instructiune

```
for (initialization; test; increment)
{
    // body
}

// initialization
// test
// increment

// body

// initialization
// test
// increment

// body
}
```

Mod de execuție

Se evaluează **Expresie_de_Initializare**

Se evaluează **Expresie_de_Testare**

Dacă **Expresie_de_Testare** este nenulă:

Se execută **Instructiune**.

Se evaluează **Expresie_de_Continuare**.

Se revine la pasul 2.

Dacă **Expresie_de_Testare** este nulă, se trece la instrucțiunea de după **for**.

Instrucțiunea **for** este echivalentă cu instrucțiunea **while**. Sintaxa descrisă mai sus este echivalentă cu:

```
for(Expresie_de_Initializare;Expresie_de_Testare;Expresie_de_Continuare )  
    Instructiune
```

```
Expresie_de_Initializare ;  
while (Expresie_de_Testare )  
{  
    Instructiune  
    Expresie_de_Continuare;  
}
```


- **Instructiune** se execută cât timp **Expresie_de_Testare** este nenulă – condiție adevărată.
- Dacă **Expresie de Testare** este de început vidă, **Instructiune** nu se execută deloc, iar **Expresie_de_Continuare** nu se mai evaluează.
- **Instructiune** poate fi orice fel de instrucțiune, dar una singură. Dacă sunt necesare mai multe instrucțiuni, se va folosi instrucțiunea compusă.
- Este necesar ca cel puțin o variabilă care apare în **Expresie_de_Testare** să-și modifice valoarea în **Instructiune** sau la evaluarea **Expresiei_de_Continuare**. Altfel se obține o buclă infinită.
- Cele trei expresii, **de_Initializare**, **_de_Testare** și **_de_Continuare** sunt separate prin caracterul **;** – obligatoriu!
- Oricare dintre cele trei expresii, **de_Initializare**, **_de_Testare** și **_de_Continuare**, eventual toate, poate să lipsească. În acest caz avem expresii vide. Dacă **Expresie_de_Testare** este vidă, rezultatul său este nenul!
- **Expresie de Initializare** se execută o singură dată. Poate să conțină și declararea unor variabile. În acest caz, variabilele vor exista numai în instrucțiunea for.

Exemplu: Suma numerelor de la 1 la n, (n număr întreg citit de la tastatură)

```
#include <iostream>
using namespace std;
int main()
{
    int n,s=0,i;
    cin>>n;
    i=1;
    while(i<=n)
    {
        s=s+i;
        i++;
    }
    cout<<s;
    return 0;
}
```

```
#include <iostream>
using namespace std;
int main()
{
    int n,s=0,i;
    cin>>n;
    for(i=1;i<=n;i++)
        s=s+i;
    cout<<s;
    return 0;
}
```

Se citește de la tastatură un număr n . Să se determine dacă numărul n este prim.

```
#include <iostream>
using namespace std;
int main()
{
    int n,i,prim=0;
    cin>>n;
    for(i=2;i<=n/2;i++)
        if(n%i==0)
            prim++;
    if(prim==0)
        cout<<"numar prim";
    else
        cout<<"numarul nu este prim";
    return 0;
}
```

Scrie un program care afișează divizorii unui număr întreg n, citit de la tastatură

```
#include <iostream>
using namespace std;
int main()
{
    int n,i,prim=0;
    cin>>n;
    for(i=2;i<=n/2;i++)
        if(n%i==0)
            cout<<i<<" ";
    return 0;
}
```