

Instrucțiunea de atribuire. Expresii

Instrucțiunea de atribuire face posibilă trimiterea unei expresii anumitei variabile:

<variabila>=<expresie>

O expresie poate conține operanzi (constante, variabile, funcții) și operatori (cei specifici limbajului).

Funcții predefinite(necesita biblioteca *math.h*):

Denumire funcție	Tip	Rezultat
pow(x,y)	double	x la puterea y
Sqrt(n)	double	Rădăcăină pătrată din n
fabs(n)	Real	Valoarea absoluta a lui n
log(n)	Real	Logaritm natural din n(n>0)
floor(n)	real	Cel mai mare intreg >=n
ceil(n)	real	Cel mai mic intreg <=r
sin(n)	double	Sinusul lui n
cos(n)	double	Cosinusul lui n
atan(n)	double	Arctangenta lui n

Exemple de expresii C++:

```
a=b=c=0; // a, b si c primesc valoarea 0 (nu se poate face atribuire multipla la declarare)
i++ //determina cresterea valorii lui i cu 1 (este echivalenta cu i=i+1)
i-- //determina scaderea valorii lui i cu 1 (este echivalenta cu i=i-1)
max=x>y?x:y // max memoreaza maximul dintre x si y
```

Citirea și scrierea datelor în C++

Citirea datelor se poate face cu cin:

cin>>nume_variabila

Exemple de citire:

```
cin>>a; //citeste valoarea variabilei a
```

```
cin>>a>>b ; //citeste valoarea variabilei a si b
```

Scrierea datelor se poate face cu cout:

Cout<<expresie

Exemple de citire:

```
cout<<a; //afiseaza valoarea variabilei a
```

```
cout<<a<<b ; //afiseaza valoarea variabilei a si b
```

```
cout<<"text de proba"; // afiseaza mesajul ,text de proba'
```

```
cout<<"Valoarea lui a este "<<a; // afisare text combinat cu afisare valoare variabila
```

STRUCTURI DE CONTROL

A. ALGORITMI CU STRUCTURĂ SECVENȚIALĂ (LINIARĂ)

Sintaxa unui astfel de algoritm este:

```
{  
  declarare variabile  
  instrucțiune1  
  instrucțiune1  
  ....  
  instrucțiune1  
}
```

În acest tip de structură instrucțiunile se execută una după alta până la sfârșitul algoritmului.

Vom porni de la următorul exemplu:

Fiind date două numere întregi notate a și b. Să se afișeze suma lor.

Algoritmul în C++ este:

```
#include<iostream.h>  
#include<conio.h>  
int main()  
{  
    int a,b,suma;  
    cin>>a;  
    cin>>b;  
    suma=a+b;  
    cout<<suma;  
    getch();  
}
```

Exemplul 2: *Se citesc valorile a două variabile a și b naturale. Să se interschimbe conținuturile lor.*

Algoritmul în C++ este:

```
#include<iostream.h>  
#include<conio.h>  
int main()  
{  
    unsigned int a,b,aux;  
    cin>>a;  
    cin>>b;  
    aux=a;  
    a=b;  
    b=aux;  
    cout<<a;  
    cout<<b;  
    getch();  
}
```

2. ALGORITMI CU STRUCTURĂ DECIZIONALĂ

Acest tip de algoritm poate lua o decizie în funcție de o condiție dată. Decizia poate fi **asupra unui singur caz** sau **între două cazuri posibile**.

a. Algoritmi cu decizie asupra unui singur caz

În acest caz dacă este îndeplinită condiția se execută o instrucțiune iar dacă nu este îndeplinită nu se acționează deloc.

Sintaxa unui astfel de algoritm este:

```
if (expresie_logica)
<instructiuni>
```

Pașii algoritmului:

- se evaluează expresia logică.
- dacă aceasta este adevărată atunci se execută instrucțiunea
- dacă aceasta nu este adevărată atunci nu se execută nimic

Exemplu:

Se citește valoarea unei variabile întregi a. Să se afișeze mesajul "Numarul este pozitiv" dacă valoarea acestei variabile este mai mare decât 0.

Algoritmul este:

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

int main()
{
    int a;
    cin>>a;
    if(a>=0)
        cout<<"Numarul este pozitiv";

    return 0;
    getch();
}
```

b. Algoritmi cu decizie între două cazuri

În acest caz dacă este îndeplinită condiția se execută o instrucțiune iar dacă nu este îndeplinită se execută o altă instrucțiune.

Sintaxa unui astfel de algoritm este:

```
if (expresie_logica)
<instructiuni1>
else
<instructiuni2>
```

Pașii algoritmului:

- se evaluează expresia logică.
- dacă aceasta este adevărată atunci se execută instrucțiunea1
- dacă aceasta nu este adevărată atunci se execută instrucțiunea2

Exemplu:

Se citește valoarea unei variabile întregi *a*. Să se afișeze mesajul "Numărul este pozitivă" dacă valoarea acestei variabile este mai mare sau egal cu 0 sau "Numărul este negativ".

Algoritmul este:

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

int main()
{
    int a;
    cin>>a;
    if(a>=0)
        cout<<"Numarul este pozitiv";
    else
        cout<<"Numarul este negativ";
    return 0;
    getch();
}
```

c. Structura de selecție - SWITCH

Acest tip de structură testează o expresie asociată, pe baza valorii acesteia transferând execuția către anumite instrucțiuni.

Forma generală este:

```
switch(optiune)
{
    case v1:s1;break;
    case v2:s2;break;
    .....
    case vm:sm;break;
    default : s;
}
```

Observație

optiune – este o expresie de tip *int* sau *char*

v1,v2 – constante către care se va face transferul execuției (trebuie să fie de același tip cu *optiune*)

Mod de lucru :

Atunci când un program întâlnește o instrucțiune switch, testează condiția asociată (expresie). Pe baza valorii întregi a acesteia, el va transfera execuția către blocul case, etichetat cu acea expresie

const corespunzatoare. În cazul când nici o expresie const nu este egală cu expresia după switch, se îndeplinește instrucțiunea din ramura default.

Exemplu :

Se citesc două numere întregi a și b. Să se calculeze în funcție de alegerea operatorului:

- Suma lor dacă s-a introdus operatorul '+'
- Diferența lor dacă s-a introdus operatorul '-'
- Produsul lor dacă s-a introdus operatorul '*'
- Catul lor dacă s-a introdus operatorul '/'

```
#include<iostream.h>
#include<conio.h>
int main()
{
    int a,b;
    char optiunea;
    cout<<"Introduceti a=";
    cin>>a;
    cout<<"Introduceti b=";
    cin>>b;
    cout<<"Optiunea=";
    cin>>optiunea;
    switch(optiunea)
    {
        case '+':
        {
            cout<<a+b;
            break;
        }
        case '-':
        {
            cout<<a-b;
            break;
        }
        case '*':
        {
            cout<<a*b;
            break;
        }
        case '/':
        {
            cout<<a/b;
            break;
        }
        default: cout<<"Optiunea gresita";
    }
    getch();
    return 0;
}
```