LIMBAJE DE PROGRAMARE

INSTRUCTIUNI / COMENZI PENTRU IMPLEMENTAREA IN LIMBAJ DE PROGRAMARE A STRUCTURII LINIARE

- În limbajul C++, structura liniară poate conține instrucțiuni de citire, scriere, declarații, atribuire, instrucțiuni compuse, instrucțiuni vide.
- Instrucțiunea compusă este o grupare de declarații și instrucțiuni cuprinse între acolade {}. Se folosește acolo unde sintaxa cere o singură instrucțiune, iar logica algoritmului cere să fie mai multe. Deși este echivalentă cu o singură instrucțiune, după instrucțiunea compusă nu se pune ; .
- Instrucțiunea vidă se folosește atunci când sintaxa limbajului C++ o cere, dar nu și logica algoritmului.
- Pentru instrucțiunea vidă se pune doar ; .
- Instrucțiunea de atribuire are forma: variabilă = expresie, unde expresie poate fi: o constantă, o expresie aritmetică care poate conține apeluri de funcție.

Exemple

```
int a, b=5; // variabila b a primit valoarea 5 la declarare
a = b + 3; // în urma atribuirii, variabila a va avea valoarea 8, b fiind 5
a = a + 2; // după atribuire, variabila a va avea valoarea 10, 8 – valoarea inițială a variabilei a + 2
```

- Dacă unei variabile de tip intreg i se va atribui o valoare reală, atunci in ea se va memora partea intreagă a valorii.
- In biblioteca **cmath** se găsesc funcții matematice (radical, modul, sin, cos etc.) și constante utile.
- De exemplu: valoarea constantei π este memorată in M_PI.

FUNCȚII UTILE DIN BIBLIOTECA CMATH

Funcție	Descriere	Exemplu
sqrt()	radical	**cout << sqrt(a+b)** afișează radicalul sumei a+b
abs()	Modul, valoare absolută	**cout << abs(a-b)** afișează modulul diferenței a-b
sin()	Sinus	**cout << sin(a*M_PI/180)** afișează sinusul unghiului **a** dat in grade
cos()	Cosinus	**cout << cos(a*M_PI/180)** afișează cosinusul unghiului **a** dat in grade
tan()	Tangenta	**cout << tan(a*M_PI/180)** afișează tangenta unghiului **a** dat in grade
floor()	Rotunjește în jos	**cout << floor(5.9)<<" "<<floor(-5.9);** afișează 5 -6
ceil()	Rotunjește în sus	**cout << ceil(5.2)<<" "<<ceil(-5.2);** afișează 6 -5
round()	Rotunjește la cel mai apropiat întreg	**cout << round(5.2)<<" "<<round(5.5);** afișează 5

TEMA

- Scrie un program care, citind lungimile catetelor - numere naturale, unui triunghi dreptunghic, calculează și afișează lungimea ipotenuzei. **Exemplu:** Pentru catetele 3 și 4, se afișează 5.
- **Rezolvare:**

a) Citești cu atenție enunțul și identifici:

- datele de intrare: lungimile catetelor și de ce tip de dată sunt – **int**;
- datele de ieșire: lungimea ipotenuzei și de ce tip de dată este – **double**.

b) Descrii in limbaj natural soluția identificată:

- Citesc lungimile catetelor,
- calculez lungimea ipotenuzei ca fiind radicalul sumei pătratelor lungimilor catetelor
- afișez rezultatul.

c) Implementezi descrierea intr-un limbaj de programare.

AFIȘAREA CU UN NUMĂR STABILIT DE ZECIMALE

- Poți afișa numerele reale cu un număr de zecimale ales de tine. Pentru aceasta, incluzi biblioteca **iomanip**, iar la tipărire folosești **setprecision**(nr) pentru a stabili numărul de cifre zecimale care se afișează. Poți combina cu **fixed** care stabilește că valoarea se va afișa cu exact numărul de cifre specificat (chiar dacă ultimele cifre sunt 0).
- **Exemplu:** Pentru declarația double a=2.34 și comanda de afișare **cout << fixed << setprecision(3) << a**, se afișează 2.340.

```
#include <iostream>
#include<cmath>
#include<iomanip>
using namespace std;

int main()
{
    int a,b;
    float c;
    cin>>a>>b;
    c=sqrt(a*a+b*b);
    cout<<fixed<<setprecision(2)<<c;
    return 0;
}
```

- Scrie un program care afișează radicalul unui număr natural **n** cu o precizie de exact două zecimale.
- **Exemplu:** Pentru **n** = 4, se afișează 2.00.

- Ionuț are o sumă de **x** lei in bancnote de un leu. Cum numărul de bancnote este mare și se încurcă in păstrarea lor, ar dori să schimbe suma in bancnote de 100 și de 10 lei. Pentru a-l convinge că operația merită făcută, scrie un program care, citind numărul **x**, afișează numărul minim de bancnote de 100, 10, 1 pe care-l va avea după schimbare. **Exemplu:** Pentru **x** = 253 se afișează 10, deoarece după schimbare va avea 2 bancnote de 100 lei, 5 de 10 lei și 3 de un leu.

```
#include <iostream>
#include<cmath>
#include<iomanip>
using namespace std;

int main()
{
    int x,nr,s,z,u;
    cin>>x;
    s=x/100;
    x=x-s*100;
    z=x/10;
    x=x-z*10;
    u=x;
    nr=s+z+u;
    cout<<nr;
    return 0;
}
```

- Marcel are **x** iepuroaice gestante. Câți iepuri va avea Marcel, știind că fiecare iepuroaică va da naștere la exact **y** pui. Scrie un program care, citind numerele **x** și **y**, va afișa numărul total de iepuri după ce toate iepuroaicele nasc.
- **Exemplu:** Pentru **x** = 4 și **y** = 2, se va afișa 12.

```
#include <iostream>
using namespace std;
int main()
{
    int x,y,s=0;
    cin>>x>>y;
    s=x+x*y;
    cout<<s;
    return 0;
}
```

- Limbajele C și C++ au fost dezvoltate de ingineri și informaticieni, fiind gandite să ofere flexibilitate in scrierea programelor. O mare parte dintre instrucțiunile introduse in aceste limbaje sunt gandite să ofere programatorului posibilitate de a scrie cat mai puțin cod pentru atingerea scopului.