**Functii de tip valoare**

Funcţiile de tip valoare calculează şi returnează o singură valoare asociată numelui funcţiei.

1. **Definirea funcţiei**

***tip\_rezultat nume\_funcţie (lista\_parametri\_formali) // antetul funcţiei***

*{*

*// declaraţii variabile locale*

*// instrucţiuni*

***return expresie\_rezultat;***

*}*

unde:

* **tip\_rezultat –** reprezintă tipul rezultatului întors de funcţie şipoate fi tip simplu predefinit (**întreg, real sau char**) sau tip **pointer/referinţă**
* **expresie\_rezultat** – trebuie să fie de acelaşi tip cu **tip\_rezultat** (altfel tipul ei va fi implicit convertit la tipul rezultatului)

**Obs.**

* Funcţia poate conţine mai multe instrucţiuni ***return expresie\_rezultat.*** La prima instrucţiune ***return*** întâlnită se iese din funcţie, revenindu-se în modulul apelant (modulul din care este apelată funcţia).
* ***lista\_parametri formali*** – conţine datele de intrare pentru subproblema rezolvată de funcţie şi este opţională. Fiecare parametru formal va fi precedat de tipul său, iar parametri vor fi separaţi prin virgulă.

1. **Apelul funcţiei**

Utilizarea unei funcţii presupune apelul ei. Apelul se face prin:

***nume\_funcţie (lista\_parametri\_actuali/efectivi)***

Lista de parametri efectivi este formată din numele parametrilor separaţi prin virgulă.

**Apelul de funcţie valoare este un operand în cadrul unei expresii**; el se inserează în locul în care este dorită valoarea produsă de funcţie. Deci poate apărea în cadrul unei expresii, într-o instrucţiune de afişare sau în membrul drept al unei atribuiri.

**Parametri efectivi/actuali trebuie să corespundă ca număr şi în ordinea definirii lor ca tip cu parametri formali din antetul funcţiei.**

**Ex.1** Să se scrie o funcţie care returnează aria unui triunghi de laturi a, b, c (cu a, b, c – numere reale).

float arie (float a, float b, float c)

{ float p, s; // variabile locala

p=(a+b+c)/2;

s=sqrt(p\*(p-a)\*(p-b)\*(p-c);

return s; // sau direct: return sqrt(p\*(p-a)\*(p-b)\*(p-c);

}

**Exemple de apel (in main() ):**

cin>>a>>b>>c; cout<<**arie(a,b,c)**; // apel utilizat intr-o afisare

**sau**

cin>>a1>>b1>>c1; s=**arie(a1,b1,c1)**; cout<<s; // apel utilizat in membrul drept al unei atribuiri

**sau**

v=**arie(3,4,5)** + **arie(6,8,10)**; // apeluri utilizate in cadrul unei expresii

**Ex.2** Să se scrie o funcţie care verifică dacă un număr natural n primit ca parametru este număr prim. Valoarea returnată va fi 1 sau 0.

int prim (int n)

{ int i;

if (n<2) return 0;

for (i=2; i\*i<=n; i++)

if (n%i==0) return 0;

return 1;

}

**Exemplu de apel:**

int main()

{int n;

cin>>n;

if (prim(n)) cout<<”DA”;

else cout<<”NU”;

return 0;

}

**Tema: 896, 897, 24, 2859, 14, 16, 904, 36, 27, 76, 113, 74 - Pbinfo**