

Fundamentele Programarii – Laborator 10

Parametri prin referinta. Tablouri ca parametri

Obiective

La finalul acestui laborator, studentul va fi capabil sa:

- intelega diferenta dintre apelul prin valoare si prin referinta;
- utilizeze parametri prin referinta in functii;
- transmite tablouri ca parametri de functie;
- modifice date in functii apelate.

Apelul prin valoare vs. apelul prin referinta

Apelul prin valoare

La apelul prin valoare, functia primeste o copie a argumentului.

```
1      void f(int x) {  
2          x = 10;  
3      }  
4  
5      int main() {  
6          int a = 5;  
7          f(a);  
8          cout << a; // 5  
9      }
```

Apelul prin referinta

La apelul prin referinta, functia lucreaza direct cu variabila originala.

```
1      void f(int &x) {  
2          x = 10;  
3      }  
4  
5      int main() {  
6          int a = 5;  
7          f(a);  
8          cout << a; // 10  
9      }
```

De ce avem nevoie de referinte?

- pentru a modifica variabilele apelantului;
- pentru a evita copieri inutile;
- pentru a lucra eficient cu structuri si tablouri.

Tablouri ca parametri

Transmiterea unui tablou

Cand un tablou este transmis unei functii, se transmite adresa primului element.

```
1         void citire(int a[], int n) {  
2             for (int i = 0; i < n; i++)  
3                 cin >> a[i];  
4         }
```

Dimensiunea tabloului nu este cunoscuta in functie, de aceea trebuie transmisa separat.

Exemplu complet

```
1         void dublare(int a[], int n) {  
2             for (int i = 0; i < n; i++)  
3                 a[i] *= 2;  
4         }  
5  
6         int main() {  
7             int v[5] = {1, 2, 3, 4, 5};  
8             dublare(v, 5);  
9             for (int i = 0; i < 5; i++)  
10                cout << v[i] << " ";  
11                return 0;  
12        }
```

Tablouri bidimensionale ca parametri

```
1         void afisare(int a[][10], int n, int m) {  
2             for (int i = 0; i < n; i++) {  
3                 for (int j = 0; j < m; j++)  
4                     cout << a[i][j] << " ";  
5                     cout << endl;  
6                 }  
7         }
```

Numarul de coloane trebuie specificat.

Exercitii propuse

Exercitii de baza

1. Scrieti o functie care interschimba valorile a doua variabile.
2. Scrieti o functie care citeste un tablou.

Exercitii aplicative

1. Scrieti o functie care sorteaza un tablou.
2. Scrieti o functie care calculeaza suma elementelor unui tablou.

Exercitii bonus

1. Scrieti o functie care returneaza minimul si maximul prin referinta.
2. Modificati o matrice folosind o functie.

Observatii

Apelul prin referinta este esential pentru programarea eficienta si pentru intelegerea pointerilor.

Laboratorul urmator: Pointeri