



7. Apel prin valoare și apel prin referință

Există două tipuri de apel al subprogramelor:

- **Apel prin valoare**
- **Apel prin referință**

7.1. Apel prin valoare – se transmite o copie a parametrului actual.

Valorile transmise la apelul unui subprogram sunt memorate în stivă. Datorită faptului că, după terminarea execuției unui subprogram, stiva este eliberată, în cazul apelului prin valoare parametrul actual **nu se modifică (se operează asupra unei copii a parametrului efectiv)**

Parametrii transmiși prin valoare se pot modifica în corpul funcției dar după terminarea apelului funcției **au aceași valoare** pe care au avut-o înainte de apel.



```
#include <iostream>
using namespace std;
int a,b;
void afis(int a, int b)
{
    a=a+1;
    b=b+4;
    cout<<a<<" "<<b<<endl;
}

int main()
{
    a=1;
    b=2;
    afis(a,b);
    cout <<a<<" " <<b<< endl;
}
```

```
#include <iostream>
using namespace std;
int a,b;
void afis(int &a, int &b)
{
    a=a+1;
    b=b+4;
    cout<<a<<" "<<b<<endl;
}

int main()
{
    a=1;
    b=2;
    afis(a,b);
    cout <<a<<" " <<b<< endl;
}
```

```
#include <iostream>
using namespace std;
int a,b;
void afis(int &m, int n)
{
    m=m+1;
    n=n+4;
    cout<<m<<" "<<n<<endl;
}

int main()
{
    a=1;
    b=2;
    cout <<a<<" " <<b<< endl;
    afis(a,b);
    cout <<a<<" " <<b;
}
```



Se va afisa:

2 6 (rezultatul apelului functiei afis)

1 2 (rezultatul de dupa apelul functiei afis)

Se va afisa:

2 6 (rezultatul apelului functiei afis)

2 6 (rezultatul de dupa apelul functiei afis)

Se va afisa:

1 2 (rezultatul inaintea apelului functiei afis)

**2 6 (rezultatul apelului functiei afis)
2 2 (rezultatul de dupa apelul functiei afis: variabila a isi pastreaza valoarea dupa apel fiind parametru transmis prin referinta pe cand variabila b revine la valoarea pe care o avea inainte de apel fiind parametru transmis prin valoare)**



7.2. Apel prin referință - se transmite adresa parametrului actual.

În cazul apelului prin referință, subprogramul, cunoscând adresa parametrului actual, **acționează direct asupra locației de memorie** indicată de aceasta, **modificând** valoarea parametrului actual.

Parametrii transmiși prin referință se folosesc pentru transmiterea de rezultate în afara funcției. Ei se pot modifica în corpul funcției dar după terminarea apelului funcției au valoarea pe care au primit-o în timpul apelului funcției.

În C++, implicit, apelul se face prin valoare. Pentru a specifica un apel prin referință, în lista parametrilor formali, numele parametrului formal va trebui precedat de simbolul **&**

**Exemplul 7.1**

```
# include <iostream>
using namespace std;
void schimba_valoare (int x, int y)
{
    int z=x;
    x = y;
    y = z;
}
void schimba_referinta (int &a, int &b)
{
    int aux=a;
    a=b;
    b=aux;
}
int main ()
{
    int M=1, N=5;
    schimba_valoare(M,N);
    cout << "M="<<M<< " " << "N="<<N<<endl;

    schimba_referinta(M,N);
    cout << "M="<<M<< " " << "N="<<N<<endl;
}
```

APEL PRIN VALOARE

APEL PRIN REFERINȚĂ

Se va afișa:
M=1 N=5

M=5 N=1