

Recursivitate

Definiție: O [functie](#) se numește [recursivă](#) dacă ea se autoapelează.

Fie următorul exemplu :Sa se calculeze n!

Varianta de program iterativă este:

```
#include <iostream>
using namespace std;
int factorial(int n)
{
    int s=1;
    for(int i=1;i<=n;i++)
        s=s*i;
    return s;
}
int main()
{
    int n;
    cout << "n=";
    cin>>n;
    cout<<"n!="<<factorial(n);
    return 0;
}
```

În cazul variantei recursive :

Se observa ca $n != (n-1) ! * n$ si se stie ca $0 != 1$. Asadar :

$$5 != 4 ! * 5$$

$$4 != 3 ! * 4$$

$$3 != 2 ! * 3$$

$$1 != 0 ! * 1$$

$$0 != 1 \text{ (prin conventie matematica)}$$

Definitia recursiva a lui $n !$ este :

$$n! = \begin{cases} (n-1)! * n, & \text{daca } n > 0 \\ 1, & \text{altfel} \end{cases}$$

Functia recursiva se va scrie astfel:

```
long factorial(int n)
{
    if(n==0)
        return 1;
    else
        return factorial(n-1)*n;
}

void main()
{
    cout<<factorial(5);
}
```

Se observa ca functia factorial se autoapeleaza. Autoapelul se realizeaza prin instructiunea `return factorial(n-1)*n`.

Reguli pentru construirea unui program recursiv:

1. Corpul unui program recursiv trebuie definit folosind doua elemente:
 - a. Cazul general al soluției (aici se execută autoapelul) – Expl: `return factorial(n-1)*n;`
 - b. Cazul de bază al soluției (aici se rezolva un caz special al problemei fără a se efectua autoapelul) – Expl: `return 1;`
2. Este obligatorie condiția de oprire a repetiției – Expl: `if(n==0)`

Observații :

1. Mecanismul recursivității înlocuiește instrucțiunile repetitive.
2. Procesul recursiv se face în 2 etape : mai întâi se descompune problema pornind de la o valoare cerută până la valoarea inițială cunoscută (adică de sus în jos), iar în etapa a doua se rezolvă problema pornind de la valoarea inițială cunoscută până la obținerea valorii cerute (de jos în sus)
3. Știm că o variabilă locală a unui subprogram poate fi definită în interiorul subprogramului și folosită doar în cadrul său. În cadrul subprogramelor recursive, la fiecare autoapel al subprogramului se face o nouă definire a variabilei locale, la n definiri făcându-se astfel n imagini ale aceleiași variabile locale, fiecare imagine având un conținut diferit. Astfel, în stivă se vor găsi toate instanțele activării subprogramului recursiv

Expl: Se citește un șir de caractere până la caracterul 0. Să se afișeze șirul inversat.

```
#include <iostream>
using namespace std;
void inv_sir()
{
    char ch;
    cin>>ch;
    if(ch!='0')
        inv_sir();
    cout<<ch;
}
int main()
{
    cout<<"Sirul=";
    inv_sir();
    return 0;
}
```

Probleme:

1. Suma primelor n numere

```
#include <iostream>
using namespace std;
int suma(int n)
{
    if(n==0) return 0;
    else
        return n+suma(n-1);
}
int main()
{
    int n;
    cout << "n=";
    cin>>n;
    cout<<"Suma primelor "<<n<<" numere este"<<suma(n);
    return 0;
}
```

2. Aflarea celui mai mare divizor comun

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

int cmmdc(int a, int b)
{
    if((a!=0)and(b!=0))
        return cmmdc(b,a%b);
    else{
        if(b==0) return a;
        if(a==0) return b;
    }
}
int main()
{
    int a,b;
    cout << "a=";
    cin>>a;
    cout << "b=";
    cin>>b;
    cout<<cmmdc(a,b);
    return 0;
}
```

3. Aflarea minimului si maximului dintr-un sir de numere

```
#include <iostream>
using namespace std;
int max(int x, int nr)
{
    if(x>nr) return x;
    else
```

```

        max(nr,x);
    }
    int min(int x, int nr)
    {
        if(x<nr) return x;
        else
            min(nr,x);

    }

    int main()
    {
        int n,nr,max1,min1;
        cout << "Cate numere introduceti=";
        cin>>n;
        cout<<"Primul numar=";
        cin>>nr;
        max1=nr;
        min1=nr;
        for(int i=2;i<=n;i++)
        {
            cout<<"nr=";
            cin>>nr;
            max1=max(max1,nr);
            min1=min(min1,nr);
        }
        cout<<"Maximul="<<max1<<endl;
        cout<<"Minimul="<<min1<<endl;
        return 0;
    }

```

4. Citire si afisare vector de numere intregi

```

#include <iostream>
using namespace std;
int a[50],i,j;
void citire(int n1)
{
    if(n1!=0)
    {
        cout<<"\na["<<j<<"]=";
        cin>>a[j++];
        citire(n1-1);
    }
}
void afisare(int n2)
{
    if(n2!=0)
    {
        cout<<"\na["<<j<<"]=";
        cout<<a[j++];
        afisare(n2-1);
    }
}

```

```

}
int main()
{
    int n;
    cout<< "Lungimea vectorului=";
    cin>>n;
    citire(n);
    afisare(n);
}

```

5. Afisare sir de caractere invers

```

#include <iostream>
using namespace std;
void invsir()
{
    char ch;
    cin>>ch;
    if(ch!='\0')
        invsir();
    cout<<ch;
}
int main()
{
    cout << "Introduceti textul=";
    invsir();
}

```