

Laborator 1
Recursivitate

1. Fie ecuația $x^2 - Sx + P = 0$ cu $S, P \in \mathbb{R}$ și x_1, x_2 rădăcinile ecuației.

Să se calculeze $S_n = x_1^n + x_2^n$, $n \in \mathbb{N}$.

Căutăm relația de recurență pentru S_n , știind că x_1 , respectiv x_2 sunt rădăcinile ecuației date și deci îndeplinesc relațiile :

$$x_1^2 - Sx_1 + P = 0$$

$$x_2^2 - Sx_2 + P = 0$$

Înmulțim aceste relații cu x_1^{n-2} și x_2^{n-2} și adunăm relațiile obținute și rezultă:

$$S_n = x_1^n + x_2^n = S * (x_1^{n-1} + x_2^{n-1}) - P * (x_1^{n-2} + x_2^{n-2}) = S * S_{n-1} - P * S_{n-2}.$$

Astfel am obținut o relație de recurență :

$$S_0 = x^0 + x^0 = 2,$$

$$S_1 = x_1 + x_2 = S,$$

$$S_n = S * S_{n-1} - P * S_{n-2}, \text{ pentru } n \geq 2.$$

```
#include<iostream.h>
using namespace std;
int n;
float s,p,r;
float suma(int n)
{
    if(n==0) return 2;
    else if(n==1) return s;
    else return(s*suma(n-1)-p*suma(n-2));
}
int main(void)
{
    cout<<"Introduceti valorile ecuatiei de gradul II "<<endl;
    cout<<"Dati s = "; cin>>s;
    cout<<"Dati p = "; cin>>p;
    cout<<" N = "; cin>>n;
    r = suma(n);
    cout<<"Valoarea lui S("<<n<<") este "<<r;
}
```

2. Să se afle elementul maxim dintr-un vector dat.

Soluția este dată de relația recurență :

$$\text{maxim}(a_1, a_2, \dots, a_n) = \text{maxim}(a_n, \text{maxim}(a_1, a_2, \dots, a_{n-1}))$$

```
#include<iostream.h>
using namespace std;
int a[100],n,i;
int max(int a,int b)
{
    if(a>b) return a;
    else return b;
}
int maxim(int a[],int n)
{
    if(n==1) return a[0];
    else return max(a[n],maxim(a,n-1));
}
int main(void)
{
    clrscr();
    cout<<"Introduceti dimensiunea sirului n = ";
    cin>>n;
    for(i=0;i<n;i++)
    {
        cout<<"a["<<i<<"]="";
        cin>>a[i];
    }
    cout<<"Elementul maxim din vector este = "<<maxim(a,n);
}
```

3. Sa se genereze toate submultimile unei multimi.

Se foloseste un vector a care indica selectia elementelor multimii:

a[i]=1 daca elem. b[i] face parte din submultime

a[i]=0,in caz contrar

Pentru generarea submultimilor vom defini functia recursiva submultime, care are un parametru i ce reprezinta pozitia elementului care se va genera.

In cazul in care s-au generat toate elementele(i=n) se tipareste submultimea generata, in caz contrar, se trece la generarea elem. urmator(aflat pe pozitia i+1).

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

int a[40],b[40],n,i;
void tipar()
{
    for(i=1;i<=n;i++)
        if(a[i]==1) cout<<b[i]<<" ";
        cout<<endl;
}
void submultime(int i)
{
    int j;
    for(j=0;j<=1;j++)
    {
        a[i]=j;
        if(i==n) tipar();
        else submultime(i+1);
    }
}
int main(void)
{
    cout<<"Dati numarul de elemente n = ";cin>>n;
    cout<<"Dati elementele multimii "<<endl;
    for(i=1;i<=n;i++) cin>>b[i];
    submultime(1);
}
```

4. Sa se transforme un numar n, dat in baza 10, intr-o alta baza b ($2 \leq b \leq 10$).

```
#include<iostream.h>
using namespace std;
int n,b;
void baza(int n)
{
    if(n<b) cout<<n;
    else
    {
        baza(n/b);
        cout<<n%b;
    }
}
int main(void)
{
    cout<<"Dati numarul in baza 10, n = ";cin>>n;
    cout<<"Dati baza in care vreti sa se transforme "; cin>>b;
```

```
cout<<n<<" in baza "<<b<<" este ";
baza(n);
}
```

5. Se citeste un numar intreg cu cel mult 255 cifre. Sa se afiseze numarul cu cifrele in ordine inversa.

```
#include<iostream.h>
using namespace std;
#include<string.h>
char n[255],i,l;
void invers(int i)
{
    if(i<1) invers(i+1);
    cout<<n[i];
}
int main(void)
{
    cout<<"Dati numarul in n = ";cin>>n;
    l=strlen(n);
    cout<<"Numarul rasturnat este ";
    invers(0);
}
```

6. Se dau doi vectori x si y cu n componente de tip cifra. Se cere sa se calculeze suma $x[0]^y[0] + x[1]^y[1] + \dots + x[n-1]^y[n-1]$.

Se vor utiliza doua functii recursive pentru calculul unui numar a la puterea b si respectiv suma componentelor unui vector.

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

int x[50],y[50],z[50],i,n;
long int putere(int a,int b)
{
    long int i,p=1;
    for(i=1;i<=b;i++) p*=a;
    return p;
}
int suma(int t[50],int i)
{
    if(i<n) return(t[i]+suma(t,i+1));
    else return 0;
}
int main(void)
{
    cout<<"Dati numarul de elemente n = ";cin>>n;
```

```
cout<<"Dati elementele primului vector X "<<endl;
for(i=0;i<n;i++) cin>>x[i];
cout<<"Dati elementele celui de-al doilea vector Y "<<endl;
for(i=0;i<n;i++) cin>>y[i];
for(i=0;i<n;i++) z[i]=z[i]+putere(x[i],y[i]);
cout<<" suma este "<<suma(z,1);
}
```

7. Sa se determine si sa afiseze toate sufixele unui cuvânt citit de la tastatura. Se numeste **sufix** al unui cuvânt, un cuvânt format din minim o litera, maxim toate pornind de la sfarsitul cuvântului.

```
#include<stdio.h>
#include<string.h>
using namespace std;

char s[50];
int n;
void afisare(char s[],int i)
{
int j;
if (i<=n-1)
{
for(j=i;j<=n-1;j++)
cout<<s[j];
cout<<endl;
afisare(s,i+1);
}
}
int main()
{
cout<<"Cuvantul=";<<cin>>s;
n=strlen(s);
afisare(s,0);
}
```

8. Scrieți definiția completă unui subprogram C++ recursiv care returnează suma elementelor unui tablou unidimensional transmis ca parametru.

```
int v[101];
int i,n;
long suma(int v[],int n)
{
if(n==0)
return 0;
else
return v[n-1]+suma(v,n-1);
}
```

9. Să se scrie o **funcție C++ recursivă** care să determine cifra maximă și cifra minimă a unui număr natural transmis ca parametru. Funcția va întoarce rezultatele prin intermediul unor parametri de ieșire.

```
void cifmaxmin (long int n, int &max1, int &min1)
{
    if(n<=9) max1=min1=n;
    else
    {
        cifmaxmin(n/10,max1,min1);
        if(n%10<min1) min1=n%10;
        if(n%10>max1) max1=n%10;
    }
}
```

10. Să se scrie o **funcție C++ recursivă** care să returneze cea mai mare cifră pară a unui număr natural transmis ca parametru.

```
int cifmaxpar(long long n)
{
    if(n<=9&& n%2)
        return -1;
    else
        if(n<=9&& n%2==0)
            return n;
    else
    {
        if (n%2==0)
        {
            int cmax=cifmaxpar(n/10);
            if(n%10>cmax)
                return n%10;
            else
                return cmax;
        }
        else
            return cifmaxpar(n/10);
    }
}
```

Probleme propuse spre rezolvare

1. Să se ruleze programele prezentate mai sus, urmărind apelurile și valorile parametrilor de apel.
2. Să se scrie un program care să calculeze al n-lea termen din șirul lui Fibonacci, care este definit recursiv astfel:
fib[1]=0;
fib[2]=1;
fib[n]=fib[n-1] + fib[n-2]
3. Să se caute o soluție nerecursivă pentru șirul lui Fibonacci.
4. Să se calculeze funcția Manna-Pnueli, dată de relația :
f(x)= | x-1, dacă x>=12
| f(f(x+2)), dacă x<12
5. Să se scrie un program recursiv care să compare două șiruri de caractere.
6. Să scrie un program recursiv care să verifice dacă două șiruri de caractere sunt **anagrame**. Două cuvinte sunt anagrame dacă conțin aceleași litere dar în ordine diferită.
Exemplu: șirul 'DARIAN' este anagrama pentru șirul 'ADRIAN'.
7. Funcție recursivă Ok care verifică dacă o valoare primită printr-un parametru x reprezintă factorialul unei valori. Funcție recursivă care citește n numere naturale și le afișează pe acelea care pot reprezenta factorialul unei valori.
8. Funcția recursivă Cif care determină cifra dominantă a unei valori primite printr-un parametru x. Cifra dominantă este prima cifră. Funcție recursivă care citește n numere naturale și le afișează pe acelea care au cifra dominantă 5.
9. Funcția recursivă Nr care determină numărul de cifre pare ale unei valori primite prin parametru x. Funcție recursivă care citește n numere naturale și afișează numărul de cifre pare al fiecărui număr citit.

Subprograme recursive – mecanism de execuție

1. Pentru definiția de mai jos a subprogramului f, ce se afișează ca urmare a apelului f(121,1)?

```
void f(long n, int i)
{
    if(n==0) cout<<i;
    if(n%3>0) f(n/3,i+1);
}
```

2. Pentru definiția alăturată a subprogramului f, ce se afișează ca urmare a apelului f(26)?

```
void f (int x)
{
    if(x>0)
    {
        cout<<'x';
        f(x/3);
        cout<<'y';
    }
}
```

3. Pentru subprogramul f definit alăturat, ce se afișează ca urmare a apelului f(15,2)?

```
void f (int n, int x)
{
    if(x>n)
        cout<<'*';
    else
    {
        f(n,x+4);
        cout<<x%10;
    }
}
```

4. Pentru definiția de mai jos a subprogramului f, ce se afișează ca urmare a apelului f(10)?

```
void f (int b)
{
    if(3<=b)
    {
        f(b-2);
        cout<<'*';
    }
    else cout<<b;
}
```


PROIECTAREA ALGORITMILOR

Laborator 1

5. Se consideră subprogramul cu definiția alăturată. Ce valoare se va afișa în urma executării instrucțiunii de mai jos?

```
cout<<f(12);
```

```
int f (int n)
{
    int c;
    if (n!=0)
    {
        if (n%2==1)
            c=1+f(n/2);
        else c=f(n/2);
        cout<<n%2;
        return c;
    }
    else return 0;
}
```

6. Se consideră subprogramul cu definiția alăturată. Ce se va afișa în urma apelului **f(12345)**?

```
void f(long int n)
{
    if (n!=0)
    {
        if (n%2==0)
            cout<<n%10;
        f(n/10);
        if (n%2!=0)
            cout<<n%10;
    }
    else cout<<endl;
}
```

7. Subprogramul recursiv alăturat este definit incomplet. Înlocuiți punctele de suspensie cu o expresie, astfel încât, în urma apelului, subprogramul f să returneze suma primelor două cifre ale numărului primit prin intermediul parametrului x?

Exemplu: în urma apelului f(2318) valoarea returnată este 5.

```
int f(int x)
{
    if (...)
        return x%10 + x/10;
    else
        return f(x/10);
}
```

8. Se consideră subprogramul recursiv alăturat, definit incomplet. Cu ce valoare trebuie înlocuite punctele de suspensie, pentru ca funcția să returneze cifra minimă a numărului natural nenul transmis prin intermediul parametrului x?

```
int Min(int x)  
{  
    int c;  
    if (x==0) return ...;  
    else {  
        c=Min(x/10);  
        if (c < x%10) return c;  
        else return x%10;  
    }  
}
```

9. Se consideră subprogramul recursiv alăturat, S, definit incomplet. Cu ce expresie pot fi înlocuite punctele de suspensie astfel încât, în urma apelului S(2), să se afișeze 3 caractere * ?

```
void S(int x)  
{  
    cout<<'*';  
    if (...) {  
        cout<<'*';  
        S(x-1);  
    }  
}
```

10. Ce afișează subprogramul F, descris alăturat, la apelul F(5);?

```
void F(int x)  
{  
    cout<<x;  
    if(x>=3)  
        F(x-2);  
}
```