

**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 :

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

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

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

**Soluție:**

```
#include<iostream>
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 s, p, n: "<<endl;
cin>>s;
cin>>p;
cin>>n;
r = suma(n);
cout<<"Valoarea lui S("<<n<<") este "<<r;
}
```

## PROIECTAREA ALGORITMILOR

### Laborator 1

Execuția programului folosind compilatorul online <https://www.jdoodle.com/online-compiler-c++/>:

```
1 #include<iostream>
2 using namespace std;
3 int n;
4 float s,p,r;
5 float suma(int n)
6 {
7     if(n==0) return 2;
8     else if(n==1) return s;
9     else return(s*suma(n-1)-p*suma(n-2));
10 }
11 int main(void)
12 {
13     cout<<"Introduceti valorile ecuatiei de gradul II s, p, n: "<<endl;
14     cin>>s;
15     cin>>p;
16     cin>>n;
17     r = suma(n);
18     cout<<"Valoarea lui S("<<n<<") este "<<r;
19 }
20
21
```

Execute Mode, Version, Inputs & Arguments

GCC 11.1.0  Interactive

Stdin Inputs

3  
2  
4

CommandLine Arguments

Result

CPU Time: 0.00 sec(s), Memory: 3796 kilobyte(s)

```
Introduceti valorile ecuatiei de gradul II s, p, n:
Valoarea lui S(4) este 17
```

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

Soluția este dată de relația recurență :

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

Soluție:

```
#include<iostream>
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)
{
    cin>>n;
for(i = 0; i < n; i++)    cin>>a[i];
cout<<"Elementul maxim din vector este = "<<maxim(a,n);
}
    
```

Executia programului folosind compilatorul online <https://www.jdoodle.com/online-compiler-c++/>:

The screenshot shows an online C++ compiler interface. At the top, there is a code editor with the following code:

```

1 #include<iostream>
2 using namespace std;
3 int a[100], n, i;
4 int max(int a, int b)
5 {
6     if(a > b) return a;
7     else return b;
8 }
9 int maxim(int a[], int n)
10 {
11     if(n == 1) return a[0];
12     else return max(a[n], maxim(a, n - 1));
13 }
14 int main(void)
15 {
16     cin>>n;
17     for(i = 0; i < n; i++)    cin>>a[i];
18     cout<<"Elementul maxim din vector este = "<<maxim(a,n);
19 }
20
21
    
```

Below the code editor, there is a control panel with the following settings:

- Compiler: GCC11.1.0
- Interactive:
- Stdin Inputs: 5, 4725 4444 333 222 111
- CommandLine Arguments: (empty)

At the bottom right, there is a blue "Execute" button and several utility icons (copy, refresh, etc.).

Below the control panel, the "Result" section shows the following output:

```

Result
CPU Time: 0.00 sec(s), Memory: 3480 kilobyte(s)
Elementul maxim din vector este = 4725
    
```

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>
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)
{
    cin>>n;
    for(i=1;i<=n;i++) cin>>b[i];
    submultime(1);
}

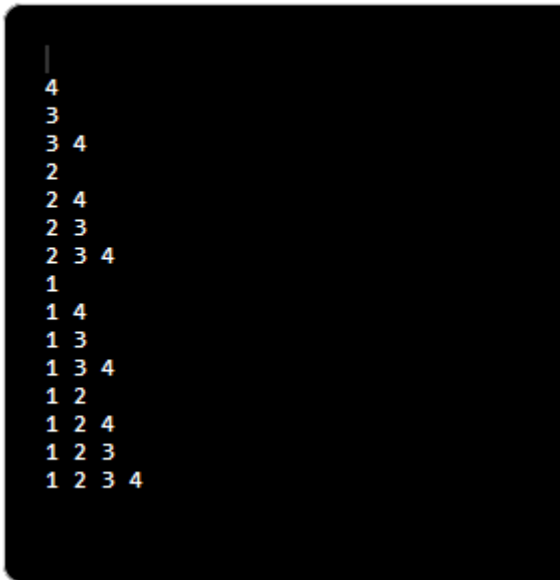
```

Executia programului folosind compilatorul online <https://www.jdoodle.com/online-compiler-c++/>:

The screenshot shows the JDoodle online compiler interface. The code editor contains the C++ code from the previous block. The 'Stdin Inputs' field is populated with the following values: 4, 1 2 3 4, 5 6 7 8, 9 10 11 12, and 13 14 15 16. The 'Execute' button is highlighted in blue at the bottom right of the interface.

Result

CPU Time: 0.00 sec(s), Memory: 3492 kilobyte(s)



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

```
#include <iostream>
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)
{
    cin>>n;
    cin>>b;
    cout<<"numarul "<<n<<" in baza "<<b<<" este ";
    baza(n);
    return 0;
}
```

## PROIECTAREA ALGORITMILOR

### Laborator 1

Executia programului folosind compilatorul online <https://www.jdoodle.com/online-compiler-c++/>:

```
1 #include <iostream>
2
3 using namespace std;
4
5 int n,b;
6 void baza(int n)
7 {
8     if(n%2) cout<<n;
9     else
10    {
11        baza(n/2);
12        cout<<n%2;
13    }
14 }
15 int main(void)
16 {
17     cin>>n;
18     cin>>b;
19     cout<<"numarul "<<n<<" in baza "<<b<<" este ";
20     baza(n);
21     return 0;
22 }
23
```

Executa Mode, Version, Inputs & Arguments

GCC 11.1.0 Interactive Stdin Inputs

CommandLine Arguments

24 2

Execute

Result

CPU Time: 0.00 sec(s), Memory: 3368 kilobyte(s)

numarul 24 in baza 2 este 11000

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

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

char n[255],i,l;
void invers(int i)
{
    if(i < l-1) invers(i+1);
    cout<<n[i];
}
int main(void)
{
    cin>>n;
    l=strlen(n);
    cout<<"Numarul initial = "<<n<<"  Numarul rasturnat este = ";
    invers(0);
}
```

## PROIECTAREA ALGORITMILOR

### Laborator 1

Executia programului folosind compilatorul online <https://www.jdoodle.com/online-compiler-c++/>:


```
1 #include <iostream>
2 #include <string.h>
3 using namespace std;
4
5 char n[255],i,i;
6 void invers(int i)
7 {
8     if(i < 1-1) invers(i+1);
9     cout<<n[i];
10 }
11 int main(void)
12 {
13     cin>>n;
14     l=strlen(n);
15     cout<<"Numarul initial = "<<n<<" Numarul rasturnat este = ";
16     invers(0);
17 }
18
19 |
```

Execute Mode, Version, Inputs & Arguments

GCC 11.1.0  Interactive Stdin Inputs

CommandLine Arguments

123456789

**Execute**   

Result

CPU Time: 0.00 sec(s), Memory: 3392 kilobyte(s)

Numarul initial = 123456789 Numarul rasturnat este = 987654321

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>
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)
{
cin>>n;
for(i=0;i<n;i++) cin>>x[i];
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);
}
```

## PROIECTAREA ALGORITMILOR

### Laborator 1

Executia programului folosind compilatorul online <https://www.jdoodle.com/online-compiler-c++/>:

```
1 #include <iostream>
2 using namespace std;
3
4 int x[50],y[50],z[50],i,n;
5 long int putere(int a,int b)
6 {
7     long int p=1;
8     for(i=1;i<=b;i++) p*=a;
9     return p;
10 }
11 int suma(int t[50],int i)
12 {
13     if(i==0) return t[i]+suma(t,i-1);
14     else return 0;
15 }
16 int main(void)
17 {
18     cin>>n;
19     for(i=0;i<n;i++) cin>>x[i];
20     for(i=0;i<n;i++) cin>>y[i];
21     for(i=0;i<n;i++) z[i]=z[i]-putere(x[i],y[i]);
22     cout<<" suma este "+suma(z,1);
23 }
24
25
26
```

Execute Mode: Version, Inputs & Arguments

GCC 11.1.0  Interactive Stdin Inputs

CommandLine Arguments

5  
1 2 3 4 5  
6 7 8 9 0

Result

CPU Time: 0.00 sec(s). Memory: 3480 kilobyte(s)

```
suma este 268834
```

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

```
#include <iostream>
#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=";>>s;
    n = strlen(s);
    afisare(s,0);
    return 0;
}
```



## PROIECTAREA ALGORITMILOR

### Laborator 1

Executia programului folosind compilatorul online <https://www.jdoodle.com/online-compiler-c++/>:

```
1 #include <iostream>
2 #include <string.h>
3 using namespace std;
4
5 char s[50];
6 int n;
7 void afisare(char s[],int i)
8 {
9     int j;
10    if (i <= n-1)
11    {
12        for(j=i; j<=n-1; j++) cout<<s[j];
13        cout<<endl;
14        afisare(s,i+1);
15    }
16 }
17 int main()
18 {
19     cout<<"Cuvantul=";<cin>>s;
20     n = strlen(s);
21     afisare(s,0);
22     return 0;
23 }
24
25
26
27
```

Execute Mode, Version, Inputs & Arguments

GCC 11.1.0  Interactive Stdin Inputs

CommandLine Arguments

programare

Execute

Result

CPU Time: 0.00 sec(s), Memory: 3400 kilobyte(s)

```
Cuvantul=programare
programare
ogramare
gramare
rasare
asare
mare
are
re
e
```

8. Scrieți definiția completă unei **functii recursive C++** 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);
}
```

Execuția programului folosind compilatorul online <https://www.jdoodle.com/online-compiler-c++/>:

```
1 #include <iostream>
2 #include <string.h>
3 using namespace std;
4
5 int v[101];
6 int i,n;
7 long suma(int v[],int n)
8 {
9     if(n==0)
10        return 0;
11     else
12        return v[n-1]+suma(v,n-1);
13 }
14
15 int main()
16 {
17     cin>>n;
18     for(int i = 0; i <= n-1; i++)
19         cin>>v[i];
20     cout<<"Suma este = "<<suma(v, n);
21     return 0;
22 }
23
24
25
26
```

Execute Mode, Version, Inputs & Arguments

GCC 11.1.0  Interactive StdIn Inputs

CommandLine Arguments

5  
1 2 3 4 5

Execute

Result  
CPU Time: 0.00 sec(s), Memory: 3488 kilobyte(s)

Suma este = 15

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;
    }
}
```

Executia programului folosind compilatorul online <https://www.jdoodle.com/online-compiler-c++/>:

```

1 #include <iostream>
2 #include<string.h>
3 using namespace std;
4
5 long long n;
6 int min1, max1;
7 void cifmaxmin (long long n, int &max1, int &min1)
8 {
9     if(n<=9) max1=min1=n;
10    else
11    {
12        cifmaxmin(n/10,max1,min1);
13        if(n%10<min1) min1=n%10;
14        if(n%10>max1) max1=n%10;
15    }
16 }
17
18 int main()
19 {
20     cin>>n;
21     cout<<"Numarul dat este = "<<n<<"\n";
22     cifmaxmin(n,max1,min1);
23     cout<<"Cifra minima este = "<<min1<<"\n";
24     cout<<"Cifra maxima este = "<<max1;
25     return 0;
26 }
27
28
29
30

```

Execute Mode, Version, Inputs & Arguments

GCC 11.1.0  Interactive Stdin Inputs

CommandLine Arguments

1234567890

**Execute**

Result

CPU Time: 0.00 sec(s), Memory: 3492 kilobyte(s)

```

Numarul dat este = 1234567890
Cifra minima este = 0
Cifra maxima este = 9

```

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);
    }
}

```




Executia programului folosind compilatorul online <https://www.jdoodle.com/online-compiler-c++/>:

```
1 #include <iostream>
2 #include <string.h>
3 using namespace std;
4
5 long long n;
6
7 int cifmaxpar(long long n)
8 {
9     if(n <= 9 && n % 2)
10         return -1;
11     else
12         if(n <= 9 && n % 2 == 0)
13             return n;
14     else
15     {
16         if (n % 2 == 0)
17         {
18             int cmax = cifmaxpar(n / 10);
19             if(n % 10 > cmax)
20                 return n % 10;
21             else
22                 return cmax;
23         }
24         else
25             return cifmaxpar(n / 10);
26     }
27 }
28
29 int main()
30 {
31     cin >> n;
32     cout << "Numarul dat este = " << n << "\n";
33     cout << "Cifra maxima para este = " << cifmaxpar(n);
34     return 0;
35 }
36
37
38
39
```

Execute Mode, Version, Inputs & Arguments

GCC 11.1.0  Interactive StdIn Inputs: 1234567890

CommandLine Arguments

**Execute**   

Result  
CPU Time: 0.00 sec(s), Memory: 3388 kilobyte(s)

```
Numarul dat este = 1234567890
Cifra maxima para este = 8
```

## 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.

**Funcții 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;  
}
```

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);
}
```

## PROIECTAREA ALGORITMILOR

### Laborator 1

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);  
}
```