

# PROIECTAREA ALGORITMILOR

**FI-AIA-1-Proiectarea Algoritmilor-2022/2023**

**Adrian Runceanu**

# Structura cursului

- 2 ore curs – titular curs  
Sef lucr.dr. Adrian RUNCEANU
- 1 ora laborator – titular aplicații practice  
Sef lucr.dr. Adrian RUNCEANU

# Forme de examinare

- Examen final – 60%
- Prezență activă la curs – 10%
- Evaluare pe parcursul semestrului a activității de laborator – 15%
- Participare activă la laborator – 15%

# Bibliografia necesară cursului:

Mihaela Runceanu, Adrian Runceanu - **STRUCTURI DE DATE ALOCATE DINAMIC. Aspecte metodice. Implementări în limbajul C++**, 2016, Editura Academica Brancusi din Targu Jiu

[https://www.researchgate.net/publication/308938197\\_STRUCTUREI\\_DE\\_DATE\\_ALOCATE\\_DINAMIC\\_Aspecte\\_metodice\\_Implementari\\_in\\_limbajul\\_C/download](https://www.researchgate.net/publication/308938197_STRUCTUREI_DE_DATE_ALOCATE_DINAMIC_Aspecte_metodice_Implementari_in_limbajul_C/download)



## Bibliografia necesară cursului:

1. Dogaru, O., *Tehnici de programare*, Editura MIRTON, Timișoara, 2002, 2004
2. Crețu, V., *Structuri de date și algoritmi*, vol.1 – *Structuri de date fundamentale*, Editura Orizonturi Universitare, Timișoara, 2000
3. Livovschi, L., Georgescu, H., *Sinteza și Analiza algoritmilor*, Editura Științifică și Enciclopedică, București, 1986
4. Wirth, N., *Algorithms and Data Structures*, Prentice Hall, Inc., Englewood, New Jersey, 1986
5. Dr. Kris Jamsa & Lars Klander, *Totul despre C și C++ - Manualul fundamental de programare în C și C++*, ed. Teora, 1999-2006

## Bibliografia necesară cursului:

6. Liviu Negrescu, *Limbajele C si C++ pentru începători*, vol. II, Limbajul C++, ed. MicroInformatica, 1995
7. A.Runceanu, **Metode si tehnici de programare – indrumar de laborator**, Editura Academica Brancusi Targu-Jiu, 2003
8. Horia Ciocârlie, Tehnici fundamentale de programare, *Ed. Orizonturi Universitare*, 2002
9. Pagina web pentru curs:  
<http://www.runceanu.ro/adrian>

## Bibliografia necesară cursului:

Referințele bibliografice nr. 1 și 7 se pot împrumuta de la Biblioteca Facultății de Inginerie, Str. Geneva nr.3, Etaj I – lângă Decanat.

1. Suport curs - varianta electronică disponibilă pe site-ul:

<http://www.runceanu.ro/adrian>

2. Îndrumar de laborator - varianta electronică disponibilă pe site pentru fiecare lucrare de laborator.

Notă: Actualizarea site-ului se face săptămânal.

# Resurse Proiectarea Algoritmilor

1. Suport curs - varianta electronică disponibilă pe: [www.runceanu.ro/adrian/](http://www.runceanu.ro/adrian/)  
Notă: Actualizarea site-ului se face saptamanal.

2. curs pe Teams (FI-AIA-1-Proiectarea Algoritmilor-2022-2023)

3. laborator pe Teams (FI-AIA-1-Proiectarea Algoritmilor-2022-2023)



# Obiectivul general al disciplinei

- Cursul de **Proiectarea Algoritmilor** adresează studenților înscriși la programul de studii Automatică și Informatică Aplicată, organizat de facultatea de Inginerie, anul 1, semestrul 2.
- **Obiectivul general al disciplinei** al cursului de **Proiectarea Algoritmilor** se refera la analiza diverșilor algoritmi pentru rezolvarea unei probleme și alegerea celui mai eficient atat din punct de vedere al memoriei utilizate, cat si al timpului de executie; proiectarea unor algoritmi eficienți pentru rezolvarea problemelor cu calculatorul.

# Obiectivele specifice

## Curs:

- Analizează un algoritm dat și stabilește complexitatea de timp pentru el.
- Alege un algoritm mai bun dintre mai mulți existenți pentru rezolvarea unei probleme.
- Înțelege, analizează și elaborează programul pentru folosirea unei stive sau cozi pentru rezolvarea unei probleme.
- Alege între o metodă iterativă și una recursivă de implementare a unui algoritm.
- Crează un graf pentru o problemă dată și-l parcurge după o metodă cunoscută
- Analizează o problemă dată, stabilește metoda în care se încadrează pentru rezolvarea sa, elaborează programul de rezolvare a problemei prin metoda aleasă.

# Obiectivele specifice

## Laborator:

✓ *Realizarea unor aplicatii de prelucrare a datelor cu ajutorul tehnicilor de programare*

✓ Mediile de dezvoltare utilizate la implementarea aplicatiile practice pot fi:

**1.MinGW** sau **CodeBlocks** (IDE-uri instalate local pe PC)

**2.www.jdoodle.com** - compilator online



# Conținutul cursului

- Pentru o bună înțelegere a noțiunilor teoretice și practice prezentate în acest curs, este necesară parcurgerea anterioară a disciplinei Programarea calculatoarelor.
- Cursul de **Proiectarea Algoritmilor** este structurat în 14 cursuri (10 capitole) și 14 laboratoare (activități practice) la care prezența va fi obligatorie.

## Conținutul cursului

- În cadrul acestui curs se vor studia metode și tehnici de programare pentru elaborarea eficientă a algoritmilor.
- Exemplificările de la curs și implementările de la laborator se vor efectua cu ajutorul limbajului de programare - **C++**.

## Capitolele cursului

- 1. Recursivitate**
- 2. Alocarea dinamică de memorie în C++**
- 3. Liste simplu și dublu înlănțuite**
- 4. Elemente de teoria grafurilor**
- 5. Algoritmi pentru prelucrarea grafurilor**
- 6. Arbori. Arbori binari**
- 7. Metoda Greedy de elaborare a algoritmilor**
- 8. Metoda Divide et Impera de elaborare a algoritmilor**
- 9. Metoda Backtracking de elaborare a algoritmilor**
- 10. Combinatorică**

## Curs 1

# Recursivitate

# Conținutul cursului

- 1. Conceptul de recursivitate**
- 2. Recursivitate directă**
- 3. Recursivitate indirectă**
- 4. Relația dintre recursivitate și iterație**
- 5. Exemple de programe recursive**

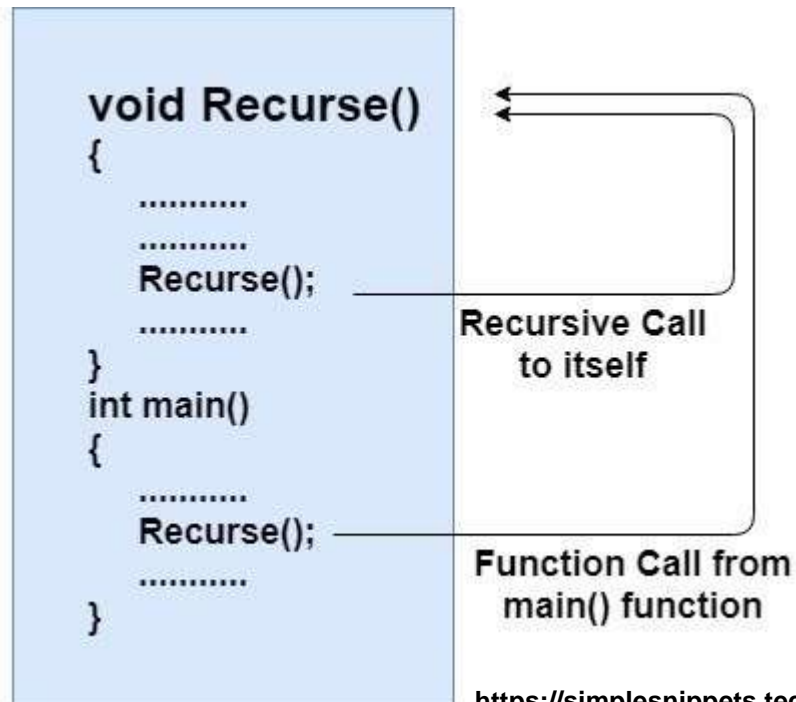


# 1. Conceptul de recursivitate

- Un obiect sau un fenomen este definit în mod **recursiv** dacă în definiția sa se face referire la el însuși.
- Conceptul de **recursivitate** oferă posibilitatea definirii unei infinități de obiecte printr-un număr finit de relații.

# 1. Conceptul de recursivitate

- O functie este recursivă atunci când executarea ei implică cel puțin încă un apel către ea însăși.*



<https://simplesnippets.tech/recursion-recursive-functions-in-cpp/>

# 1. Conceptul de recursivitate

Tipuri de recursivitate:

- 1. Recursivitate directă* – apelul recursiv se face chiar din funcția invocată.
- 2. Recursivitate indirectă (mutuală)* – apelul recursiv se realizează prin intermediul mai multor funcții care se apelează circular.

## Exemplul 1

### Definirea numerelor naturale:

- 1 este număr natural
- succesorul unui număr natural este un număr natural

Se presupune cunoscută *definiția succesorelui unui număr: acel număr obținut din numărul dat prin adăugarea unei unități.*

## Exemplul 2

Algoritm de calcul pentru factorialul unui număr  $n$ . (notatie  $n!$ ):

- dacă  $n = 0$  atunci  $n! = 1$
- dacă  $n > 0$  atunci  $n! = n * (n-1)!$

Astfel spus, *factorialul unui număr  $n > 0$  se obține prin înmulțirea numărului cu factorialul predecesorului.*

# Conținutul cursului

1. Conceptul de recursivitate
- 2. Recursivitate directă**
3. Recursivitate indirectă
4. Relatia dintre recursivitate si iteratie
5. Exemple de programe recursive

## Recursivitate directă

- În limbajul C++ funcțiile se pot apela pe ele însele, adică sunt **direct recursive**.
- Pentru o funcționare corectă (din punct de vedere logic), **apelul recursiv trebuie să fie condiționat de o decizie** care, la un moment dat în cursul execuției, **să împiedice continuarea apelurilor recursive și să permită astfel revenirea din sirul de apeluri**.

## Recursivitate directă

- *Lipsa acestei conditii* sau programarea ei gresită va conduce la executarea unui sir de apeluri a cărui terminare nu mai este controlată prin program si care, la epuizarea resurselor sistemului, va provoca o eroare de executie:

***Stack overflow (Depășirea stivei de date)***

```
int functie()  
{  
    return functie();  
}
```



## Exemplu

```
void p (listă de parametri){
```

```
lista variabile locale
```

```
...
```

```
    p(listă de parametri);
```

```
}
```

```
if (conditie) p(listă de parametri) ...
sau:
while (conditie)
{ ...p(listă de parametri)... }
sau:
do { ... p(listă de parametri)... } while
(conditie)
```

- **Conditia** care trebuie testată este specifică problemei de rezolvat.
- *Programatorul trebuie să o identifice în fiecare situație concretă și, pe baza ei, să redacteze corect apelul recursiv.*
- Revenirea din apeluri se face în ordine inversă. <sup>25</sup>

# Conținutul cursului

1. Conceptul de recursivitate
2. Recursivitate directă
3. Recursivitate indirectă
4. Relatia dintre recursivitate si iteratie
5. Exemple de programe recursive

## Recursivitate indirectă

- Un subprogram S, în corpul căruia apar apeluri la S (la el însuși) se numește subprogram **direct recursiv** iar un subprogram P, pentru care există un subprogram Q, astfel încât P face apeluri la Q, iar Q conține apelul la P se numește subprogram **indirect recursiv**.
- În acest ultim caz, subprogramele P și Q se mai numesc și **mutual recursive**.

## Recursivitate indirectă

*Funcție direct recursivă*

```
funcția S;  
{  
    ...  
    S; // apel la funcția S  
    ...  
}
```

*Funcții mutual recursive*

```
funcția P;  
{  
    ...  
    Q; // apel la funcția Q  
    ...  
}  
funcția Q;  
{  
    ...  
    P; // apelul funcției P  
    ...  
}
```

# Conținutul cursului

1. Conceptul de recursivitate
2. Recursivitate directă
3. Recursivitate indirectă
4. **Relatia dintre recursivitate si iteratie**
5. Exemple de programe recursive

# Relația dintre recursivitate și iterație - Comparație

## Iterația

- execuția repetată a unei secvențe de instrucțiuni
- o nouă iterație se execută doar în urma evaluării unei condiții (**la început sau sfârșit**)
- fiecare iterație se execută până la capăt și apoi se trece, eventual, la o nouă iterație
- se recomandă atunci când algoritmul de calcul este exprimat printr-o formulă iterativă

## Recursivitatea

- execuția repetată a unei funcții
- un nou apel recursiv se execută tot în urma evaluării unei condiții (**pe parcurs**)
- funcția recursivă se apelează din nou, înainte de terminarea apelului precedent
- se recomandă doar atunci când problema este prin definiție recursivă (recursivitatea consumă resurse în exces)

# Conținutul cursului

- 1. Conceptul de recursivitate**
- 2. Recursivitate directă**
- 3. Recursivitate indirectă**
- 4. Relatia dintre recursivitate si iteratie**
- 5. Exemple de programe recursive**

# Probleme rezolvate

1. Se dau doua numere intregi  $a$  si  $b$  si se cere sa se calculeze cel mai mare divizor comun. (**Algoritmul lui EUCLID – prin împărțiri repetate**).

Formulara recursivă, în cuvinte, a algoritmului:

- *Dacă unul dintre numere este zero, c.m.m.d.c. al lor este celălalt număr.*
- *Dacă nici unul dintre numere nu este zero, atunci c.m.m.d.c. nu se modifică dacă se înlocuieste unul dintre numere cu restul împărțirii sale cu celălalt.*



# Probleme rezolvate

Algoritmul poate fi implementat sub forma următoarei funcții recursive:

```
int cmmdc (int n, int m)  
{  
    if (n==0) return m;  
    else return cmmdc(n, m % n);  
}
```

# Probleme rezolvate

Codul sursa al implementarii (**varianta prin scaderi succesive**) este urmatorul:

```
#include<iostream>
using namespace std;
int cmmdc(int a,int b)
{
    if(a==b) return a;
    else if(a>b) return cmmdc(a-b, b);
        else return cmmdc(a, b-a);
}
```

```
int a, b;
```

```
int main(void)
```

```
{
```

```
    cin>>a>>b;
```

```
    cout<<"C.m.m.d.c. "<<a<<" "<<b<<" este  
    "<<cmmdc(a,b)<<endl;
```

```
    return 0;
```

```
}
```

Executia programului folosind compilatorul online

<https://www.jdoodle.com/online-compiler-c++/>:

The screenshot shows the JDoodle online C++ compiler IDE. The browser address bar displays `jdoodle.com/online-compiler-c++/`. The page header includes the JDoodle logo, navigation icons, and a "Sign In" button. A sponsored banner for DigitalOcean is visible below the header. The main heading is "Online C++ Compiler IDE".

The code editor contains the following C++ code:

```
1 #include<iostream>
2 using namespace std;
3 int cmmdc(int a,int b)
4 {
5     if(a==b) return a;
6     else if(a>b) return cmmdc(a-b, b);
7     else return cmmdc(a, b-a);
8 }
9 int a, b;
10
11 int main(void)
12 {
13     cin>>a>>b;
14     cout<<"C.m.m.d.c. "<<a<<","<<b<<" este "<<cmmdc(a,b)<<endl;
15     return 0;
16 }
17
```

Below the code editor, the "Execute Mode, Version, Inputs & Arguments" section is visible. It includes a dropdown menu for the compiler version (GCC 9.1.0), an "Interactive" checkbox, and a "Stdin Inputs" text area containing "246 573". There is also a "CommandLine Arguments" text area. A blue "Execute" button is located below these fields.

The "Result" section shows the output of the program: "C.m.m.d.c. 246,573 este 3". It also displays performance metrics: "CPU Time: 0.00 sec(s), Memory: 3380 kilobyte(s)" and "compiled and executed in 1.97 sec(s)".

# Probleme rezolvate

2. Să se calculeze suma primelor  $n$  numere naturale.

Soluția este dată de relația de recurență:

$$\begin{aligned} \text{suma}(1, 2, \dots, n) &= \\ \text{suma}(n, \text{suma}(1, 2, \dots, n-1)) \end{aligned}$$

Condiția de ieșire din recursivitate:

Dacă  $n=1$ , atunci suma este 1

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

Executia programului folosind compilatorul online

<https://www.jdoodle.com/online-compiler-c++/>:

The screenshot shows the JDoodle website's online C++ compiler interface. At the top, there's a navigation bar with the JDoodle logo, a 'Sign In' button, and a sponsored banner for DigitalOcean. The main heading is 'Online C++ Compiler IDE'. Below this, there's a code editor with the following C++ code:

```
1 #include<iostream>
2 using namespace std;
3 long int suma(long int i)
4 {
5     if(i==1) return 1;
6     else return suma(i-1)+i;
7 }
8 long int n;
9 int main(void)
10 {
11     cin>>n;
12     cout<<"Suma primelor "<<n<<" numere este "<<suma(n)<<endl;
13     return 0;
14 }
15
16
```

Below the code editor, there's a configuration section titled 'Execute Mode, Version, Inputs & Arguments'. It includes a dropdown menu for the compiler version (GCC 9.1.0), an 'Interactive' checkbox, and a 'Stdin Inputs' text area containing the number '10'. There's also a 'CommandLine Arguments' text area. A blue 'Execute' button is prominently displayed.

Below the configuration section, the 'Result' is shown. It indicates 'CPU Time: 0.00 sec(s), Memory: 3320 kilobyte(s)' and 'compiled and executed in 3.279 sec(s)'. The output of the program is displayed in a black box with white text: 'Suma primelor 10 numere este 55'.

# Probleme rezolvate

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

Soluția este dată de relația de recurență:

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

Condiția de ieșire din recursivitate:

Dacă  $n=1$ , atunci maximul este primul element  $a[1]$



```
#include<iostream>
using namespace std;
int a[100],n,i;
int max(int x, int y)
{
    if(x > y) return x;
    else return y;
}

int maxim(int a[ ],int n)
{
    if(n==1) return a[1];
    else return
    max(a[n],maxim(a,n-
1));
}
```

```
int main(void)
{
    cin>>n;
    for(i=1;i<=n;i++) cin>>a[i];
    cout<<"Elementul maxim din
vector este = "<<maxim(a,n);
    return 0;
}
```

# Executia programului folosind compilatorul online

<https://www.jdoodle.com/online-compiler-c++/>:

The screenshot displays the JDoodle online C++ compiler interface. At the top, the browser address bar shows the URL `jdoodle.com/online-compiler-c++/`. The JDoodle logo and a "Sign In" button are visible in the header. A sponsored banner for DigitalOcean is present below the header.

### Online C++ Compiler IDE

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

Execute Mode, Version, Inputs & Arguments

GCC 9.1.0  Interactive Stdin Inputs

CommandLine Arguments:

Stdin Inputs: 5  
12 3 56 -90 34

**Execute** ...

**Result**  
CPU Time: 0.00 sec(s), Memory: 3200 kilobyte(s) compiled and executed in 1.514 sec(s)

```
Elementul maxim din vector este = 56
```

# Probleme rezolvate

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)
{
    // Dati numarul in baza 10, n=
    cin>>n;
    // Dati baza in care vreti sa se
    // transforme
    cin>>b;
    cout<<n<<" in baza "<<b<<"
    este ";
    baza(n);
    return 0;
}
```

# Executia programului folosind compilatorul online

<https://www.jdoodle.com/online-compiler-c++/>:

The screenshot displays the JDoodle Online C++ Compiler IDE interface. At the top, the browser address bar shows the URL `jdoodle.com/online-compiler-c++/`. Below the browser, the page title is "Online C++ Compiler IDE".

The main area contains a code editor with the following C++ code:

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

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

- Execute Mode: GCC 9.1.0
- Interactive:
- Stdin Inputs: 24 2
- CommandLine Arguments: (empty field)

A blue "Execute" button is visible, along with a menu icon (three dots) and a refresh icon (circular arrow).

The "Result" section shows the output of the program:

```
24 in baza 2 este 11000
```

Additional information at the bottom of the result section includes: "CPU Time: 0.00 sec(s), Memory: 3280 kilobyte(s)" and "compiled and executed in 3.965 sec(s)".

# Probleme rezolvate

5. Se citește un număr întreg ca un șir de caractere cu cel mult 255 cifre.

Să se afișeze numărul cu cifrele în ordine inversă.

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

```
char n[255],i,l;
```

```
void invers(int i)
{
    if(i<l) invers(i+1);
    cout<<n[i];
}
```

```
int main(void)
{
    cin>>n;
    l=strlen(n);
    cout<<"Numarul rasturnat
este ";
    invers(0);
    return 0;
}
```

# Executia programului folosind compilatorul online

<https://www.jdoodle.com/online-compiler-c++/>:

The screenshot displays the JDoodle Online C++ Compiler IDE interface. At the top, the browser address bar shows the URL `jdoodle.com/online-compiler-c++/`. Below the browser, the page title is "Online C++ Compiler IDE".

The main area contains a code editor with the following C++ code:

```
1 #include<iostream>
2 #include<string.h>
3 using namespace std;
4
5 char n[255],i,l;
6
7 void invers(int i)
8 {
9     if(i<l) invers(i+1);
10    cout<<n[i];
11 }
12
13 int main(void)
14 {
15     cin>>n;
16     l=strlen(n);
17     cout<<"Numarul rasturnat este ";
18     invers(0);
19     return 0;
20 }
```

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

- A dropdown menu for the compiler version, currently set to "GCC 9.1.0".
- An "Interactive" checkbox, which is currently unchecked.
- A "Stdin Inputs" text area containing the value "12345".
- A "CommandLine Arguments" text area, which is currently empty.
- An "Execute" button with a play icon.
- Two additional buttons: a three-dot menu and a full-screen icon.

Below the control panel, the "Result" section shows the output of the program:

CPU Time: 0.00 sec(s), Memory: 3436 kilobyte(s) compiled and executed in 1.443 sec(s)

```
Numarul rasturnat este-54321
```



# Probleme rezolvate

## 6. Suma puterilor rădăcinilor

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:

$$x_1^2 - Sx_1 + P = 0 \quad | * x_1^{n-2}$$

$$x_2^2 - Sx_2 + P = 0 \quad | * x_2^{n-2}$$

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

$$\begin{aligned} S_n &= x_1^n + x_2^n = \\ &= S * (x_1^{n-1} + x_2^{n-1}) - P * (x_1^{n-2} + x_2^{n-2}) = \\ &= S * S_{n-1} - P * S_{n-2} \end{aligned}$$

Astfel am obținut o relație de recurență:

$$S_0 = x_1^1 + x_2^1 = 1 + 1 = 2, \text{ pentru } n=0$$

$$S_1 = x_1^1 + x_2^1 = S, \text{ pentru } n=1$$

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

```
#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)
{
    // Introduceti valorile
    // ecuatiei de gradul II
    // Dati s =
    cin>>s;
    // Dati p =
    cin>>p;
    // N =
    cin>>n;
    r = suma(n);
    cout<<"Valoarea lui
    S("<<n<<" este "<<r;
```

# Executia programului folosind compilatorul online

<https://www.jdoodle.com/online-compiler-c++/>:

The screenshot displays the JDoodle online C++ compiler interface. At the top, the browser address bar shows the URL `jdoodle.com/online-compiler-c++/`. Below the browser, the page title is "Online C++ Compiler IDE".

The main area contains a code editor with the following C++ code:

```
1 #include<iostream>
2
3 using namespace std;
4 int n;
5 float s,p,r;
6
7 float suma(int n)
8 {
9     if(n==0) return 2;
10    else if(n==1) return s;
11    else return(s*suma(n-1)-p*suma(n-2));
12 }
13 int main(void)
14 {
15     // Introduceti valorile ecuatiei de gradul II
16     // Dati s =
17     cin>>s;
18     // Dati p =
19     cin>>p;
20     // N =
21     cin>>n;
22     r = suma(n);
23     cout<<"Valoarea lui S("<<n<<") este "<<r;
24 }
```

Below the code editor, there is a control panel for execution. It includes a dropdown menu for the compiler version (GCC 9.1.0), an "Interactive" checkbox, and a "Stdin Inputs" text area containing "4 5 2". There is also a "CommandLine Arguments" text area. A blue "Execute" button is prominently displayed.

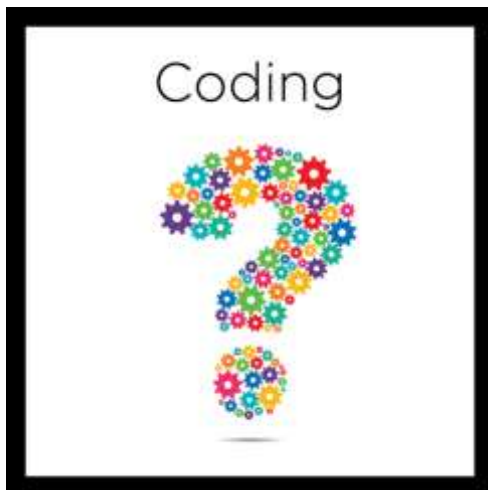
The "Result" section shows the output of the program: "Valoarea lui S(2) este 6". It also indicates the execution statistics: "CPU Time: 0.00 sec(s), Memory: 3588 kilobyte(s)" and "compiled and executed in 2.423 sec(s)".

## Probleme propuse spre rezolvare

1. Să se scrie un program care să calculeze al n-lea termen din **șirul lui Fibonacci**, care este definit recursiv astfel:
  - $\text{fib}[1]=0$
  - $\text{fib}[2]=1$
  - $\text{fib}[n]=\text{fib}[n-1] + \text{fib}[n-2]$ , pentru  $n>2$
2. Să se caute o soluție nerecursivă pentru **șirul lui Fibonacci**.

# Test Grila

Aflati ce valoare se afiseaza in secventele de cod urmatoare.



## Aflati ce valoare se afiseaza

1. Funcția **rec** are definiția alăturată. Ce valoare are `rec(3)`? Dar `rec(10)`?

```
int rec(int a)
{
    if( a == 0 )    return 0;
    else    return rec( a - 1 ) + 2;
}
```



## Aflati ce valoare se afiseaza

2. Funcția **rec** are definiția alăturată. Ce valoare are `rec(7)`? Dar `rec(100)`?

```
int rec(int a)
{
    if( a%6 == 0 )      return a;
    else return rec( a - 1 );
}
```

## Aflati ce valoare se afiseaza

3. Se consideră funcția **rec**, definită mai jos. Ce valoare are `rec(1)`? Dar `rec(20)`?

```
int rec(int n)
{
    if( n == 0 )    return 0;
    return 1 + rec( n / 2 );
}
```

## Aflati ce valoare se afiseaza

4. Funcția **rec** este definită alăturat. Ce valoare va avea `rec(7)`? Dar `rec(10)`?

```
int rec(int i)
{
    if ( i > 12 )    return 1;
    else return     1 + rec( i + 2 );
}
```

## Aflati ce valoare se afiseaza

5. Funcția **rec** este definită alăturat. Ce valoare are `rec(4,8)`? Dar `rec(5,10)`?

```
int rec(int x, int y)
{
    if( x <= y )    return 1 + rec( x + 1, y );
    return 0;
}
```

## Referinte bibliografice

1. Adrian Runceanu, ***Metode si tehnici de programare – indrumar de laborator***, Editura Academica Brancusi Targu-Jiu, 2003
  2. Octavian Dogaru, ***Tehnici de programare***, Editura MIRTON, Timișoara, 2002, 2004
- Referințele bibliografice nr. 1 și 2 se pot împrumuta de la Biblioteca Facultății de Inginerie, Str. Geneva nr.3, Etaj I – lângă Decanat.

# Întrebări?