

Ministerul Educației și Cercetării al Republicii Moldova

CENTRUL DE EXCELENȚĂ ÎN INFORMATICĂ  
ȘI TEHNOLOGII INFORMAȚIONALE

Disciplina: Programarea structurată  
Lucrare individuală nr. 1

*Executat:* Pricop Maxim

*Grupa:* P-2413

*Verificat:* Natalia Gairunova

Chișinău, 2024

## Partea I – Programe Liniare

Eu sunt numărul 26 în catalog, ceea ce înseamnă că trebuie să rezolv problema 26.

### Sarcină:

Se consideră cunoscute laturile unui dreptunghi. Sa se scrie un program in C++ care calculează aria și perimetrul acestui dreptunghi.

### Rezolvare:

```
void program1() {
    float lungime, latime;
    cout << "Introdu lungimea dreptunghiului: "; cin >> lungime;
    cout << "Introdu latimea dreptunghiului: "; cin >> latime;

    if (lungime <= 0 || latime <= 0) {
        cout << "Unul din numerele introduse este mai mic sau egal ca 0. Este imposibil de a efectua calculul." << endl;
        return;
    };

    float perimetru = 2 * (lungime + latime);
    float aria = lungime * latime;

    cout << "Perimetrul dreptunghiului este: " << perimetru << endl;
    cout << "Aria dreptunghiului este: " << aria << endl;
};
```

### Terminal:

```
(max@nurofen)-[~/.../Programing/Scoala/Programare Structurata/Lucrare Individuala]
$ cd "/home/max/Desktop/Programing/Scoala/Programare Structurata/Lucrare Individuala/" && g++ lucrare_individuala_1.cpp -o lucrare_individuala_1
1 && "/home/max/Desktop/Programing/Scoala/Programare Structurata/Lucrare Individuala/"lucrare_individuala_1
— Program nr. 1 —
Introdu lungimea dreptunghiului: 420
Introdu latimea dreptunghiului: -69
Unul din numerele introduse este mai mic sau egal ca 0. Este imposibil de a efectua calculul.

(max@nurofen)-[~/.../Programing/Scoala/Programare Structurata/Lucrare Individuala]
$ cd "/home/max/Desktop/Programing/Scoala/Programare Structurata/Lucrare Individuala/" && g++ lucrare_individuala_1.cpp -o lucrare_individuala_1
1 && "/home/max/Desktop/Programing/Scoala/Programare Structurata/Lucrare Individuala/"lucrare_individuala_1
— Program nr. 1 —
Introdu lungimea dreptunghiului: 6.9
Introdu latimea dreptunghiului: 4.2
Perimetrul dreptunghiului este: 22.2
Aria dreptunghiului este: 28.98

(max@nurofen)-[~/.../Programing/Scoala/Programare Structurata/Lucrare Individuala]
$ cd "/home/max/Desktop/Programing/Scoala/Programare Structurata/Lucrare Individuala/" && g++ lucrare_individuala_1.cpp -o lucrare_individuala_1
1 && "/home/max/Desktop/Programing/Scoala/Programare Structurata/Lucrare Individuala/"lucrare_individuala_1
— Program nr. 1 —
Introdu lungimea dreptunghiului: 6
Introdu latimea dreptunghiului: 7
Perimetrul dreptunghiului este: 26
Aria dreptunghiului este: 42
```

## Partea II – Instrucțiuni de Selecție

Eu sunt numărul 26 în catalog, ceea ce înseamnă că trebuie să rezolv problema 9.

### Sarcină:

Sa se alcatuiasca programul pentru functia data utilizind instructiunea IF.

$$F(x)=\begin{cases} x+2x^2+6, & x < -4 \\ \frac{2x(x^2+1)}{x+1}, & -4 \leq x \leq 4 \\ 2x^3+(x-11)-7x^2, & x > 4 \end{cases}$$

### Rezolvare:

```
void program2() {
    float x, y;
    cout << "Introdu numarul: "; cin >> x;

    if (x < -4) y = x + 2 * pow(x, 2) + 6;
    else if (-4 <= x && x <= 4) y = (2 * x * (pow(x, 2) + 1)) / (x + 1);
    else y = 2 * pow(x, 3) + (x - 11) - 7 * pow(x, 2);

    cout << "Rezultatul este: " << y << endl;
};
```

### Terminal:

```
(max@nurofen)-[~/.../Programing/Scoala/Programare Structurata/Lucrare Individuala]
$ cd "/home/max/Desktop/Programing/Scoala/Programare Structurata/Lucrare Individuala/" && g++ lucrare_individuala_1.cpp -o lucrare_individuala_1
1 && "/home/max/Desktop/Programing/Scoala/Programare Structurata/Lucrare Individuala/"lucrare_individuala_1
— Program nr. 2 —
Introdu numarul: -6
Rezultatul este: 72
```

```
(max@nurofen)-[~/.../Programing/Scoala/Programare Structurata/Lucrare Individuala]
$ cd "/home/max/Desktop/Programing/Scoala/Programare Structurata/Lucrare Individuala/" && g++ lucrare_individuala_1.cpp -o lucrare_individuala_1
1 && "/home/max/Desktop/Programing/Scoala/Programare Structurata/Lucrare Individuala/"lucrare_individuala_1
— Program nr. 2 —
Introdu numarul: -2
Rezultatul este: 20
```

```
(max@nurofen)-[~/.../Programing/Scoala/Programare Structurata/Lucrare Individuala]
$ cd "/home/max/Desktop/Programing/Scoala/Programare Structurata/Lucrare Individuala/" && g++ lucrare_individuala_1.cpp -o lucrare_individuala_1
1 && "/home/max/Desktop/Programing/Scoala/Programare Structurata/Lucrare Individuala/"lucrare_individuala_1
— Program nr. 2 —
Introdu numarul: 7
Rezultatul este: 339
```

## Partea III – Instrucțiuni Repetitive

Eu sunt numărul 26 în catalog, ceea ce înseamnă că trebuie să rezolv problema 26.

### Sarcină:

Sa se scrie un program in C++ care afla ultimul element n din sirul Fibonacci si il afiseaza la ecran, unde n este introdus de la tastatura.

### Rezolvare:

```
void program3() {
    int n;
    cout << "Introdu cate numere Fibonacci vrei sa fie calculate: "; cin >> n;

    if (n <= 0) {
        cout << "Trebuie sa introduci un numar mai mare ca 0." << endl;
        return;
    } else if (n == 1) {
        cout << "Nr. 1 din secventa Fibonacci este 0." << endl;
        return;
    } else if (n == 2) {
        cout << "Nr. 2 din secventa Fibonacci este 1." << endl;
        return;
    }

    int i = 3, firstNumber = 0, secondNumber = 1, lastNumber;

    // Metoda 1 - For Loop
    for (i; i <= n; ++i) {
        lastNumber = firstNumber + secondNumber;

        firstNumber = secondNumber;
        secondNumber = lastNumber;
    };

    // Metoda 2 - While Loop
    // while (i <= n) {
    //     lastNumber = firstNumber + secondNumber;

    //     firstNumber = secondNumber;
    //     secondNumber = lastNumber;
    //     i++;
    // };

    // Metoda 3 - Do While Loop
    // do {
    //     lastNumber = firstNumber + secondNumber;

    //     firstNumber = secondNumber;
    //     secondNumber = lastNumber;
    //     i++;
    // } while (i <= n);

    cout << "Nr. " << n << " din secventa Fibonacci este " << lastNumber << endl;
};
```

# Lucrare Individuală 1

## Terminal:

```
(max@nurofen)-[~/../Programing/Scoala/Programare Structurata/Lucrare Individuala]
$ cd "/home/max/Desktop/Programing/Scoala/Programare Structurata/Lucrare Individuala/" && g++ lucrare_individuala_1.cpp -o lucrare_individuala_1
1 && "/home/max/Desktop/Programing/Scoala/Programare Structurata/Lucrare Individuala/"lucrare_individuala_1
— Program nr. 3 —
Introdu cate numere Fibonacci vrei sa fie calculate: -1
Trebuie sa introduci un numar mai mare ca 0.
```

```
(max@nurofen)-[~/../Programing/Scoala/Programare Structurata/Lucrare Individuala]
$ cd "/home/max/Desktop/Programing/Scoala/Programare Structurata/Lucrare Individuala/" && g++ lucrare_individuala_1.cpp -o lucrare_individuala_1
1 && "/home/max/Desktop/Programing/Scoala/Programare Structurata/Lucrare Individuala/"lucrare_individuala_1
— Program nr. 3 —
Introdu cate numere Fibonacci vrei sa fie calculate: 0
Trebuie sa introduci un numar mai mare ca 0.
```

```
(max@nurofen)-[~/../Programing/Scoala/Programare Structurata/Lucrare Individuala]
$ cd "/home/max/Desktop/Programing/Scoala/Programare Structurata/Lucrare Individuala/" && g++ lucrare_individuala_1.cpp -o lucrare_individuala_1
1 && "/home/max/Desktop/Programing/Scoala/Programare Structurata/Lucrare Individuala/"lucrare_individuala_1
— Program nr. 3 —
Introdu cate numere Fibonacci vrei sa fie calculate: 1
Nr. 1 din secventa Fibonacci este 0.
```

```
(max@nurofen)-[~/../Programing/Scoala/Programare Structurata/Lucrare Individuala]
$ cd "/home/max/Desktop/Programing/Scoala/Programare Structurata/Lucrare Individuala/" && g++ lucrare_individuala_1.cpp -o lucrare_individuala_1
1 && "/home/max/Desktop/Programing/Scoala/Programare Structurata/Lucrare Individuala/"lucrare_individuala_1
— Program nr. 3 —
Introdu cate numere Fibonacci vrei sa fie calculate: 2
Nr. 2 din secventa Fibonacci este 1.
```

```
(max@nurofen)-[~/../Programing/Scoala/Programare Structurata/Lucrare Individuala]
$ cd "/home/max/Desktop/Programing/Scoala/Programare Structurata/Lucrare Individuala/" && g++ lucrare_individuala_1.cpp -o lucrare_individuala_1
1 && "/home/max/Desktop/Programing/Scoala/Programare Structurata/Lucrare Individuala/"lucrare_individuala_1
— Program nr. 3 —
Introdu cate numere Fibonacci vrei sa fie calculate: 3
Nr. 3 din secventa Fibonacci este 1
```

```
(max@nurofen)-[~/../Programing/Scoala/Programare Structurata/Lucrare Individuala]
$ cd "/home/max/Desktop/Programing/Scoala/Programare Structurata/Lucrare Individuala/" && g++ lucrare_individuala_1.cpp -o lucrare_individuala_1
1 && "/home/max/Desktop/Programing/Scoala/Programare Structurata/Lucrare Individuala/"lucrare_individuala_1
— Program nr. 3 —
Introdu cate numere Fibonacci vrei sa fie calculate: 4
Nr. 4 din secventa Fibonacci este 2
```

```
(max@nurofen)-[~/../Programing/Scoala/Programare Structurata/Lucrare Individuala]
$ cd "/home/max/Desktop/Programing/Scoala/Programare Structurata/Lucrare Individuala/" && g++ lucrare_individuala_1.cpp -o lucrare_individuala_1
1 && "/home/max/Desktop/Programing/Scoala/Programare Structurata/Lucrare Individuala/"lucrare_individuala_1
— Program nr. 3 —
Introdu cate numere Fibonacci vrei sa fie calculate: 7
Nr. 7 din secventa Fibonacci este 8
```

## **Concluzie**

Principala concluzie a lucrării este că, prin intermediul conceptelor fundamentale ale limbajului C++, se pot realiza aplicații simple, dar eficiente, pentru a rezolva diverse probleme. Concluzia principală a lucrării evidențiază importanța abilităților de bază în programare, cum ar fi validarea inputului, utilizarea condiționalelor și implementarea programelor repetitive, pentru a oferi soluții flexibile și corecte.

Printre principalele puncte discutate se numără realizarea unui program pentru calculul ariei și perimetrului unui dreptunghi, un program pentru evaluarea unei funcții matematice în funcție de valori condiționate ale inputului, și al treilea pentru generarea unei secvențe Fibonacci folosind mai multe tipuri de bucle. Aceste exemple au demonstrat cum se pot implementa metode de calcul și logică în C++, ilustrând flexibilitatea limbajului.

Rezultatele principale arată că fiecare program și-a atins scopul, oferind rezultate corecte și gestionând inputurile nevalide, acolo unde a fost cazul. În mod specific, fiecare program a oferit o metodă clară și eficientă pentru calculul rezultatului cerut, validând astfel teza că printr-o programare bine structurată se pot obține soluții clare.

**Siteuri Folosite**

W3Schools: [https://www.w3schools.com/cpp/cpp\\_for\\_loop.asp](https://www.w3schools.com/cpp/cpp_for_loop.asp)