**MINISTERUL EDUCAȚIEI ȘI CERCETĂRII AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică şi Microelectronică**

**Departamentul Informatică şi Ingineria Sistemelor**

**RAPORT**

Lucrare de laborator nr. 3

la cursul „***Programarea Calculatoarelor***”

**Tema: Prelucrarea tablourilor unidimensionale în limbajul C**

A efectuat :  **St. gr. CR-221FR Serba Cristina**

**A verificat: Asis.univ. Toma Olga**

**Chișinău 2022**

**Cuprins**

[3](#_Toc1216030167)

[INTRODUCERE 3](#_Toc462410381)

[REALIZAREA PRACTICĂ A SARCINII DE LUCRU 4](#_Toc679806610)

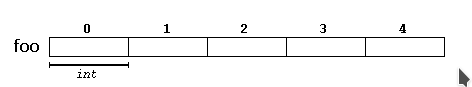
[CONCLUZII 7](#_Toc35190996)

[BIBLIOGRAFIE 8](#_Toc1213899961)

# 

# INTRODUCERE

Un vector (sau șir sau tablou unidimensional) reprezintă o serie de elemente de același tip, memorate în locații de memorie succesive care pot fi referite individual prin adăugarea unui index la un identificator unic. Aceasta înseamnă că, de exemplu, cinci valori de tip int pot fi declarate ca un vector fără a fi necesară declararea a cinci variabile diferite (fiecare cu propriul identificator). Intr-adevăr, folosind un vector, cele cinci valori de tip int sunt stocate în locații de memorie continuă și toate cinci pot fi accesate folosind același identificator, dar cu index-ul potrivit. De exemplu, un vector conținând cinci valori întregi de tip int denumit foo ar putea fi reprezentat astfel:



**Fig 1 *Un vector foo***

unde fiecare spațiu liber reprezintă un element al șirului. În acest caz, acestea sunt valori de tip int. Aceste elemente sunt numerotate de la 0 la 4, primul fiind pe poziția 0 și 4 indicându-l pe ultimul; în C++, primul element dintr-un vector este totdeauna numerotat cu zero (nu cu unu), indiferent de lungimea vectorului.

Lucrarea dată urmează să fie efectuată prin parcurgerea elementelor vectorului și efectuarea unor instrucțiuni de control pentru manipularea rezultatului final.

# REALIZAREA PRACTICĂ A SARCINII DE LUCRU

**Sarcina lucrării de laborator (varianta 2):**

Pentru tabloul unidimensional dat format din n elemente de tip real să se determine valoarea minimală dintre elementele tabloului şi poziţiile elementelor cu această valoare, precum şi produsul tuturor elementelor pozitive în tablou.

#include <stdio.h>

int main()

{

    int n;

    float A[255];

    printf("dati # de elemente:\n");

    scanf("%d", &n);

    printf("dati elementele tabloului:\n");

    for(int i = 0; i < n; i++)

        scanf("%f", &A[i]);

    float minimal = A[0], produs = 1;

        int indice;

    for(int i = 0; i < n; i++)

    {

        if(A[i] < minimal) minimal = A[i];

        if(A[i] > 0) produs \*= A[i];

    }

    printf("pozitiile cu cea mai mica valoare a tabloului sunt:");

    for(int i = 0; i < n; i++)

        if(A[i] == minimal) printf(" %d", i+1);

    printf("\nprodusul elementelor pozitive din tablou este: %f\n", produs);

    return 0;

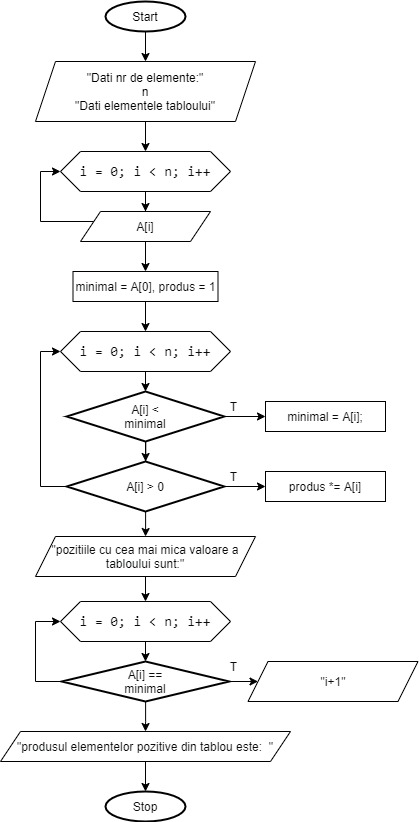
}

Programul citește numărul de elemente din matrice de la utilizator folosind scanf, apoi citește elementele matricei folosind un ciclu for și scanf.

Programul inițializează o variabilă minimală la primul element al matricei și o variabilă produs la 1.

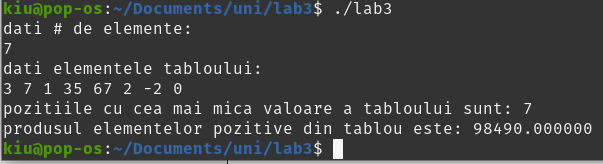
Apoi iterează prin matrice folosind un ciclu și actualizează variabila minimală la valoarea minimă întâlnită până acum; produs - la produsul tuturor elementelor pozitive întâlnite până acum.

Programul imprimă apoi indicii elementelor din matrice care au valoarea minimă folosind o altă buclă. În cele din urmă, imprimă produsul elementelor pozitive din matrice.



**Fig.2 Schema bloc a programului**

Figura 2 reprezintă schema bloc a programului



# CONCLUZII

Scopul lucrării date este de a prelucra un tabel unidimensional pentru a determina unele concluzii despre elementele sale. Conform condiției, rezultatul final constă din elementul minim și produsul elementelor pozitive. Pentru îndeplinirea sarcinii, am creat variabile de control cu care este parcurs fiecare element al tabelului, și ale căror valori se modifică în funcție de îndeplinirea condițiilor stabilite.

Este necesară inițializarea unei variabile de control cu unul dintre elementele vectorului, pentru a putea, ulterior, să fie comparate fiecare dintre elementele rămase cu acesta și să fie determinată valoarea maximă a tabloului.

De asemenea, am folosit un singur ciclu pentru a putea realiza ambele condiții.

Ținând cont de aceste lucruri, am remarcat ca prelucrarea tabelelor unidimensionale se face doar prin ciclurile care accesează fiecare element.

# BIBLIOGRAFIE

[***https://cplusplus.com/doc/tutorial-ro/arrays/***](https://cplusplus.com/doc/tutorial-ro/arrays/)

[***https://app.diagrams.net/***](https://app.diagrams.net/)