

Universitatea Tehnică a Moldovei
Facultatea *Calculatoare, Informatică și Microelectronică*
Specialitatea *Tehnologii Informaționale*



Raport

la lucrarea de laborator nr. 6

Tema: “*Subprograme ale utilizatorului. Funcții.*”

Disciplina: “Programarea Calculatorului”

Varianta 4

A efectuat:

Student grupa TI-231 FR

Apareci Aurica

A verificat:

Asistent universitar

Mantaluță Marius

Chișinău 2023

Cuprins

1. Cadrul teoretic	3
2. Schema bloc.....	3
3. Listingul programului	4
4. Testarea aplicației.....	5
5. Concluzii.....	6

1. Cadrul teoretic

Tema: Subprograme ale utilizatorului. Funcții.

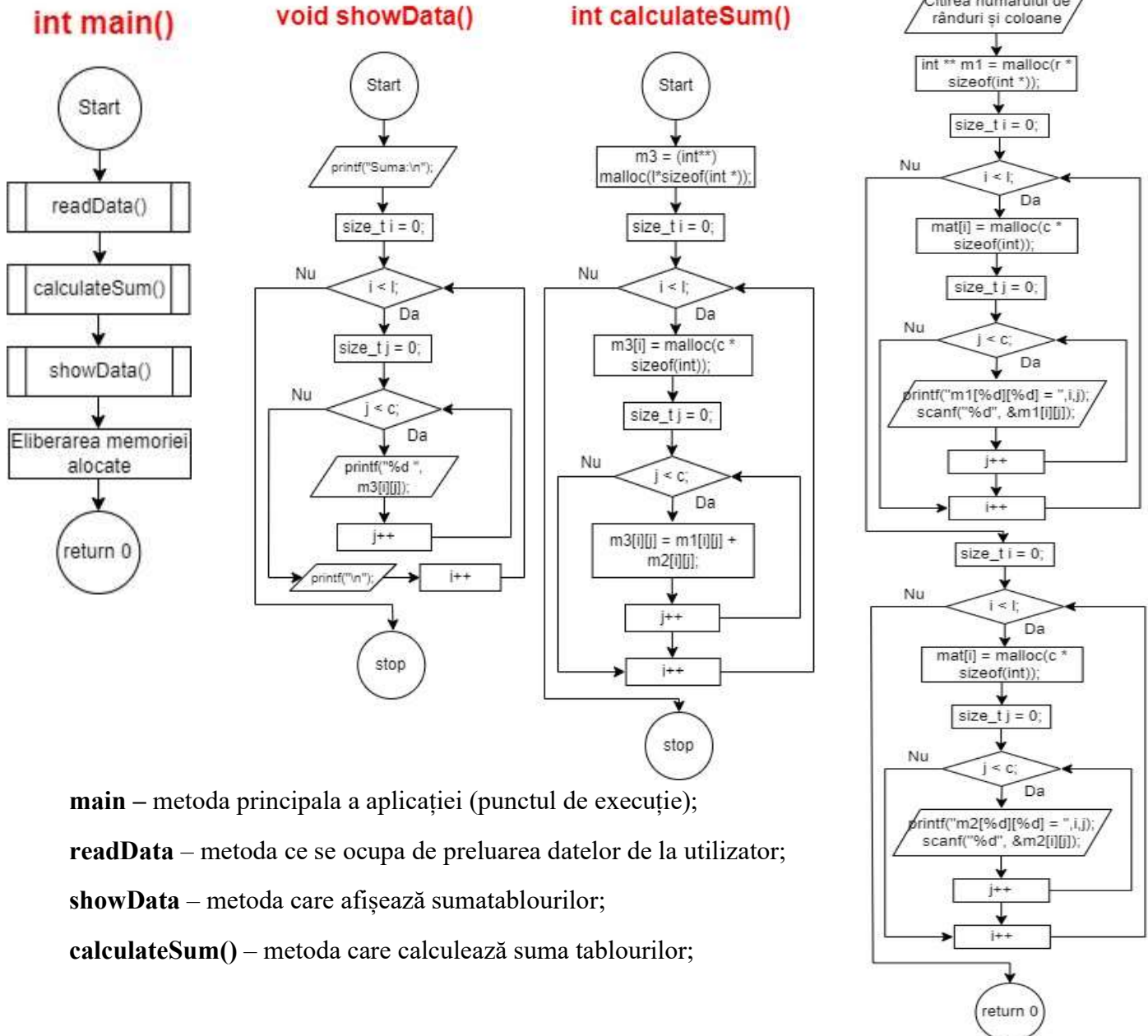
Scopul lucrării: Însușirea, folosirea și obținerea deprinderii practice de elaborare și depanare a subprogramelor(funcțiilor).

Sarcina: Să se elaboreze schema bloc și programul cu funcția necesară pentru soluționarea următoarelor probleme:

sumTab()

De calculat suma a două tablouri de aceeași dimenisune.

2. Schema bloc



3. Listingul programului

```
#include <stdio.h>
#include <stdlib.h>
```

```
int l, c;
int ** m1, ** m2, ** m3;
```

```
void readData()
```

```
{
    printf("Introduceti numarul de linii: ");
    scanf("%d", &l);
    printf("Introduceti numarul de coloane: ");
    scanf("%d", &c);
    m1 = (int **)malloc(l * sizeof(int *));
    printf("Introduceti elementele primului tablou:\n");
    for (size_t i = 0; i < l; i++)
    {
        m1[i] = (int *)malloc(c * sizeof(int));
        for (size_t j = 0; j < c; j++)
        {
            printf("m1[%d][%d] = ", i, j);
            scanf("%d", &m1[i][j]);
        }
    }
    m2 = (int **)malloc(l * sizeof(int *));
    printf("Introduceti elementele celui de-al doilea tablou:\n");
    for (size_t i = 0; i < l; i++)
    {
        m2[i] = (int *)malloc(c * sizeof(int));
        for (size_t j = 0; j < c; j++)
        {
            printf("m2[%d][%d] = ", i, j);
            scanf("%d", &m2[i][j]);
        }
    }
}
```

```
int calculateSum()
```

```
{
    m3 = (int**)malloc(l*sizeof(int *));
    for (size_t i = 0; i < l; i++)
    {
        m3[i]=(int*)malloc(c*sizeof(int));
        for (size_t j = 0; j < c; j++)
        {
            m3[i][j] = m1[i][j] + m2[i][j];
        }
    }
}
```

```
void showData()
```

```
{
    printf("Suma celor doua tablouri
    este:\n");
    for (size_t i = 0; i < l; i++)
    {
        for (size_t j = 0; j < c; j++)
        {
            printf("%d ", m3[i][j]);
        }
        printf("\n");
    }
}
```

```

int main()
{
    readData();
    calculateSum();
    showData();

    for (size_t i = 0; i < l; i++)
    {
        free(m1[i]);
        free(m2[i]);
        free(m3[i]);
    }
    free(m1);
    free(m2);
    free(m3);
    return 0;
}

```

4. Testarea aplicației

Nr.	Input	Output
1.	<p>Introduceti numarul de linii: 3 Introduceti numarul de coloane: 3 Introduceti elementele primului tablou:</p> <pre> m1[0][0] = 1 m1[0][1] = 2 m1[0][2] = 3 m1[1][0] = 4 m1[1][1] = 5 m1[1][2] = 6 m1[2][0] = 7 m1[2][1] = 8 m1[2][2] = 9 </pre> <p>Introduceti elementele celui de-al doilea tablou:</p> <pre> m2[0][0] = 9 m2[0][1] = 8 m2[0][2] = 7 m2[1][0] = 6 m2[1][1] = 5 m2[1][2] = 4 m2[2][0] = 3 m2[2][1] = 2 m2[2][2] = 1 </pre>	<p>Suma celor doua tablouri este:</p> <pre> 10 10 10 10 10 10 10 10 10 </pre>

5. Concluzii

În concluzie, această lucrare de laborator a abordat utilizarea subprogramelor pentru calcularea sumei a două tablouri de aceeași dimensiune, datele cărora sunt introduse de către utilizator. Programul a fost dezvoltat în limbajul C, iar funcțiile definite au permis realizarea eficientă a acestei sarcini. În cadrul laboratorului, pentru realizarea sarcinii:

1. Am citit numărul de linii și coloane, precum și elementele acestora de la utilizator.
2. Am calculat suma celor două tablouri, înlocuind fiecare element al tabloului rezultat cu suma elementelor corespondente din cele două tablouri de intrare.
3. Am afișat tabloul rezultat, reprezentând suma celor două tablouri inițiale.

Programul a fost proiectat pentru a gestiona tablouri de orice dimensiune specificată de utilizator, asigurând astfel flexibilitate și adaptabilitate. La final am eliberat memoria alocată pentru cele trei tablouri pentru a preveni scurgeri de memorie. Acest laborator a demonstrat beneficiile utilizării subprogramelor în programarea C pentru a organiza mai eficient codul și pentru a realiza operații complexe, precum adunarea a două tablouri, într-un mod clar și structurat.