**Universitatea Tehnică a Moldovei**

**Facultatea *Calculatoare, Informatică și Microelectronică***

**Specialitatea *Tehnologii Informaționale***

A blue and black logo

Description automatically generated

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** 2](#_Toc146231008)

[**2.** **Schema bloc** 3](#_Toc146231009)

[**3.** **Listingul programului** 4](#_Toc146231010)

[**4.** **Testarea aplicaței** 5](#_Toc146231011)

[**5.** **Concluzii** 5](#_Toc146231012)

# **Cadrul teoretic**

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

**Scopul lucrării:** Însuşirea, folosirea şi obţinerea deprinderii practice de elaborare şi depanare a subprogramelor(funcţiilor).

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

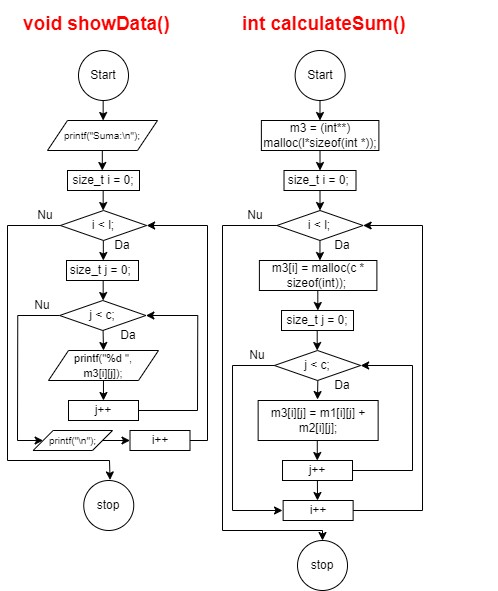


A diagram of a block diagram

Description automatically generated

# **Schema bloc**

A diagram of a data flow

Description automatically generated

**main –** metoda principala a aplicației (punctul de execuție);

**readData** – metoda ce se ocupa de preluarea datelor de la utilizator;

**showData** – metoda care afișează sumatablourilor;

**calculateSum()** – metoda care calculează suma tablourilor;

# **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() | **void showData()** |
| {  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];  }  }  } | {  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;  } | |

# **Testarea aplicaței**

|  |  |  |
| --- | --- | --- |
| Nr. | Input | Output |
| 1. |  |  |

# **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.