

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



# Raport

**la lucrarea de laborator nr. 5**

**Tema: “*Prelucrarea caracterelor și a șirurilor de caractere*”**

**Disciplina: “Programarea Calculatorului”**

Varianta 4

**A efectuat:**

Student grupa TI-231 FR

Apareci Aurica

**A verificat:**

Asistent universitar

Mantaluță Marius

**Chișinău 2023**

# Cuprins

<b>1. Cadrul teoretic.....</b>	<b>3</b>
<b>2. Listingul programului.....</b>	<b>3</b>
<b>3. Schema bloc.....</b>	<b>3</b>
<b>4. Testarea aplicației.....</b>	<b>4</b>
<b>5. Concluzii.....</b>	<b>4</b>

# 1. Cadrul teoretic

**Tema:** Prelucrarea caracterelor și a șirurilor de caractere.

**Scopul lucrării:** Însușirea, folosirea și obținerea deprinderii practice de elaborare și depanare a programelor cu algoritmilor de prelucrare a caracterelor și a șirurilor de caractere

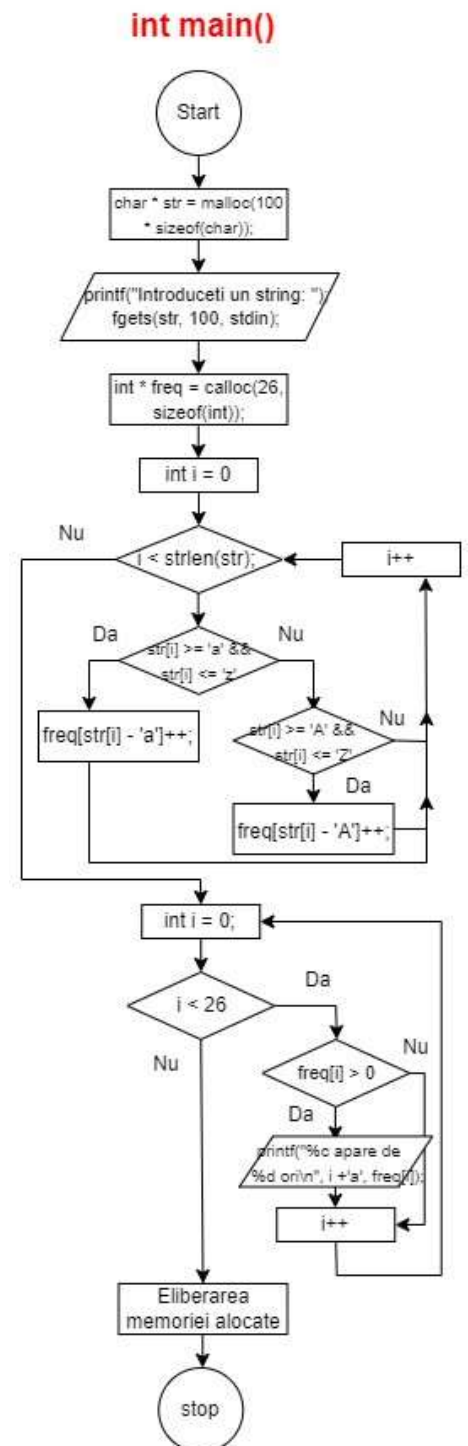
**Sarcina:** Să se elaboreze schema bloc și programul care ar manipula șirurile de caractere conform condițiilor din **Tabelul 1**: Să se determine frecvența repetării fiecărei litere.

## 2. Listingul programului

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

int main()
{
    char * str = malloc(100 * sizeof(char));
    printf("Introduceti un string: ");
    fgets(str, 100, stdin);
    int * freq = calloc(26, sizeof(int));
    for(int i = 0; i < strlen(str); i++)
    {
        if(str[i] >= 'a' && str[i] <= 'z')
        {
            freq[str[i] - 'a']++;
        }
        else if(str[i] >= 'A' && str[i] <= 'Z')
        {
            freq[str[i] - 'A']++;
        }
    }
    for(int i = 0; i < 26; i++)
    {
        if(freq[i] > 0)
        {
            printf("%c apare de %d ori\n", i + 'a',
            freq[i]);
        }
    }
    free(str);
    free(freq);
}
```

## 3. Schema bloc



## 4. Testarea aplicației

Nr. caz	Input	Output
1.	<code>Introduceti un string: Programarea Calculatorului</code>	<code>a apare de 5 ori c apare de 2 ori e apare de 1 ori g apare de 1 ori i apare de 1 ori l apare de 3 ori m apare de 1 ori o apare de 2 ori p apare de 1 ori r apare de 4 ori t apare de 1 ori u apare de 3 ori</code>
2.	<code>Introduceti un string: UTM FCIM TI 231 FR</code>	<code>c apare de 1 ori f apare de 2 ori i apare de 2 ori m apare de 2 ori r apare de 1 ori t apare de 2 ori u apare de 1 ori</code>
3.	<code>Introduceti un string: ABCabcAbC</code>	<code>a apare de 3 ori b apare de 3 ori c apare de 3 ori</code>

## 5. Concluzii

În cadrul lucrării de laborator am implementat un program în limbajul C care primește un șir de caractere de la utilizator și determină frecvența repetării fiecărei litere din alfabetul englez. Acest lucru a fost realizat prin intermediul următorilor pași:

1. Am alocat dinamic o zonă de memorie pentru stocarea șirului de caractere introdus de utilizator, cu o capacitate maximă de 100 de caractere și am folosit funcția `fgets` pentru a permite utilizatorului să introducă un șir de caractere de la tastatură.
2. Am alocat dinamic un vector de frecvențe pentru cele 26 de litere ale alfabetului englez, inițializând toate frecvențele cu zero utilizând funcția `calloc`.
3. Am parcurs fiecare caracter din șirul introdus de utilizator și am determinat dacă acesta este o literă mică ('a'-'z') sau o literă mare ('A'-'Z'). Pentru fiecare literă găsită, am incrementat frecvența corespunzătoare din vectorul de frecvențe.
4. Am afișat rezultatele și am eliberat memoria alocată dinamic pentru șirul de caractere și vectorul de frecvențe folosind funcția `free` pentru a preveni scurgerile de memorie.

În concluzie, acest program a fost capabil să analizeze și să determine cu succes frecvența repetării fiecărei litere dintr-un șir de caractere introdus de utilizator. Această abilitate poate fi utilă într-o varietate de aplicații, cum ar fi analiza textelor, procesarea datelor sau generarea de statistici privind utilizarea literelor într-un text dat.