****

**MINISTERUL EDUCAȚIEI, CULTURII ȘI CERCETĂRII**

**AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică şi Microelectronică**

**Departamentul Informatică şi Ingineria Sistemelor**

**Calancea Cătălin**

**MI-222**

**Raport**

**pentru lucrarea de laborator Nr.1**

***la cursul de “Structuri de date şi algoritmi”***

Verificat:

**Guțu Maria,** *doctor, conf. univ.*

Departamentul Informatică şi IS,

Facultatea FCIM, UTM

**Chișinău – 2023**

**Scopul lucrării:** Studierea posibilităţilor şi mijloacelor limbajului C pentru programarea algoritmilor cu structură ramificată şi ciclică la prelucrarea tablourilor 1D de tip Char, manipulând datele tablourilor prin elaborarea funcțiilor de către utilizator și transmiterea datelor către funcții prin pointeri.

**Nr. variantei:** 3

**Condiția problemei ( sarcinii de lucru ) :**

Elaborați un program C care va crea un meniu recursiv. Acesta trebuie să cuprindă următoarele funcții în C cu apelare ulterioară ale acestora în funcția main

1. Citește șirul de caractere de la tastatură;

2. Afișează șirul de caractere;

3. Calculează și afișează lungimea șirului;

4. Să se modifice toate caracterele ultimului cuvânt în litere mari;

5. să se identifice cea mai mare propoziție din șir;

6. să se numere câte cuvinte există în șir;

7. să se înlocuiască toate combinațiile de caractere ‘ma’ întâlnite în șir cu caracterele ‘\*\*’.

0.Iesirea din meniu.

**Codul programului:**

#include <stdio.h>

#include <ctype.h>

#include <string.h>

#define MAX\_STRING\_LENGTH 1000

#define MAX\_SENTENCE\_LENGTH 100

void readString(char str[]);

void printString(const char str[]);

int stringLength(const char str[]);

void LastWord(char str[]);

void LongestSentence(const char str[], char longestSentence[]);

int countWords(const char str[]);

void Stars(char str[]);

void menu();

int main() {

menu();

return 0;

}

void readString(char str[]) {

fgets(str, MAX\_STRING\_LENGTH, stdin);

if (str[strlen(str)-1] == '\n') {

str[strlen(str)-1] = '\0';

}

}

void printString(const char str[]) {

printf("Sirul de caractere este: %s\n", str);

}

int stringLength(const char str[]) {

return strlen(str);

}

void LastWord(char str[]) {

int length = stringLength(str);

int i, j;

for (i = length - 1; i >= 0; i--) {

if (isspace(str[i])) {

break;

}

}

for (j = i + 1; j < length; j++) {

str[j] = toupper(str[j]);

}

}

void LongestSentence(const char str[], char longestSentence[]) {

int maxLength = 0;

int length = 0;

int start = 0;

int end = 0;

int i;

int n = stringLength(str);

for (i = 0; i < n; i++) {

if (str[i] == '.' || str[i] == '!' || str[i] == '?') {

length = i - start + 1;

if (length > maxLength) {

maxLength = length;

end = i;

}

start = i + 1;

}

}

strncpy(longestSentence, &str[end-maxLength+1], maxLength);

longestSentence[maxLength] = '\0';

}

int countWords(const char str[]) {

int count = 0;

int i;

int n = stringLength(str);

for (i = 0; i < n; i++) {

if (isspace(str[i])) {

count++;

}

}

return count + 1;

}

void Stars(char str[]) {

char \*p = strstr(str, "ma");

while (p != NULL) {

strncpy(p, "\*\*", 2);

p = strstr(p+2, "ma");

}

}

void menu() {

int choice;

char str[MAX\_STRING\_LENGTH];

char longestSentence[MAX\_SENTENCE\_LENGTH];

do {

printf("Meniu:\n");

printf("1. Citeste sirul de caractere\n");

printf("2. Afiseaza sirul de caractere\n");

printf("3. Calculeaza si afiseaza lungimea sirului\n");

printf("4. Modifica toate caracterele ultimului cuvant in litere mari\n");

printf("5. Identifica cea mai mare propozitie din sir\n");

printf("6. Numara cate cuvinte exista in sir\n");

printf("7. Inlocuieste toate combinatiile de caractere 'ma' cu '\*\*'\n");

printf("0. Iesire\n");

printf("Alegeti o optiune: ");

scanf("%d", &choice);

getchar(); // Consumam caracterul newline

switch (choice) {

case 1:

printf("Introduceti sirul de caractere: ");

readString(str);

break;

case 2:

printString(str);

break;

case 3:

printf("Lungimea sirului este: %d\n", stringLength(str));

break;

case 4:

LastWord(str);

printf("Sirul de caractere cu ultimul cuvant in litere mari este: %s\n", str);

break;

case 5:

LongestSentence(str, longestSentence);

printf("Cea mai mare propozitie din sir este: %s\n", longestSentence);

break;

case 6:

printf("Numarul de cuvinte din sir este: %d\n", countWords(str));

break;

case 7:

Stars(str);

printf("Sirul de caractere cu 'ma' inlocuit cu '\*\*' este: %s\n", str);

break;

case 0:

printf("La revedere!\n");

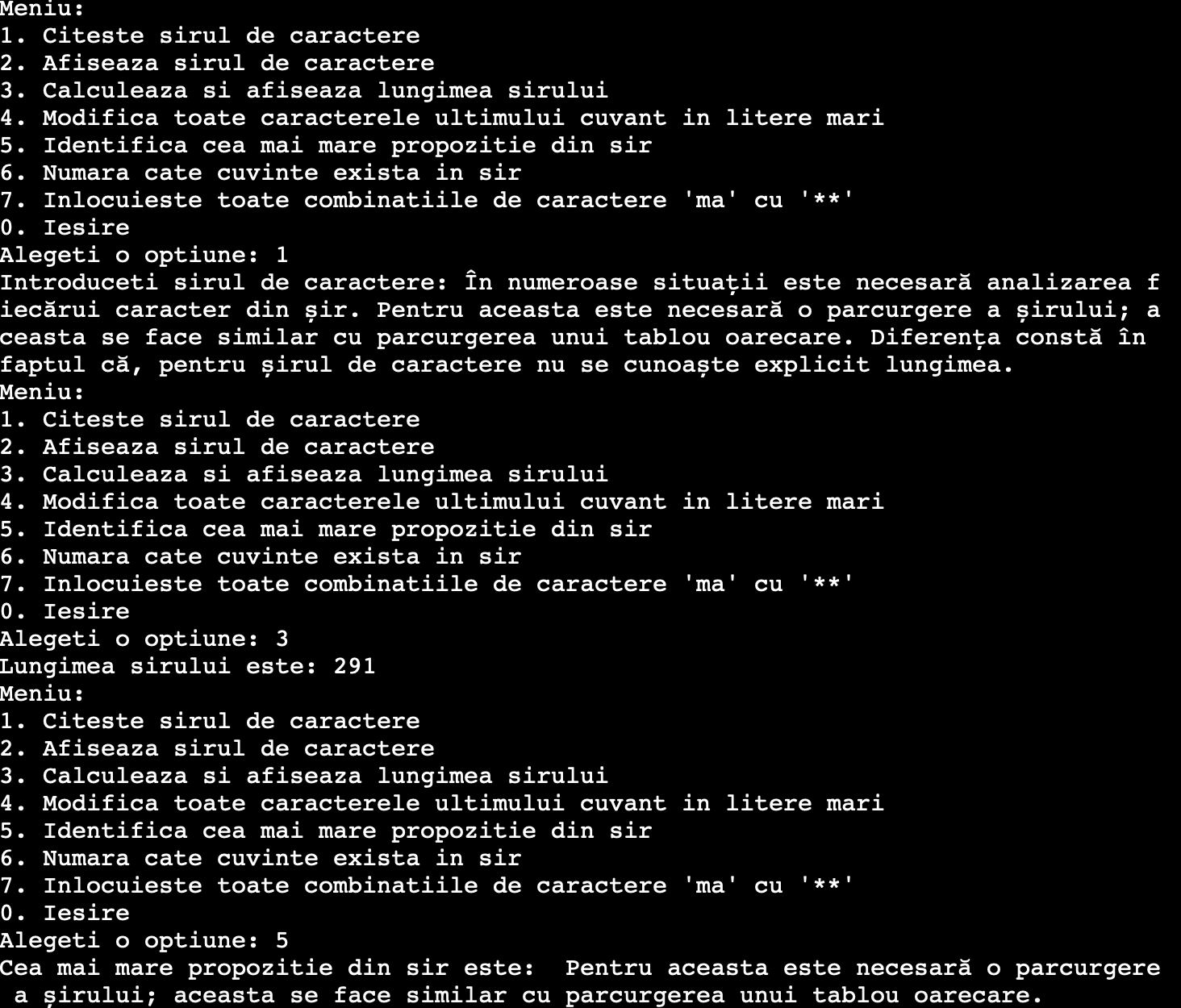
break;

default:

printf("Optiune invalida!\n");

}

} while (choice != 0);}



**Concluzie:**În urma executarii lucrării de laborator nr.1 , am facut cunostinta mai aprofundata cu functiile si cu recursivitatea si iteravitatea , de asemenea am creat un meniu recursiv. Am efectuat citirea si unele modficari asupra charurilor .