

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



# Raport

**la lucrarea de laborator nr. 4**

**Tema: “*Algoritmi de prelucrare a fișierelor textuale*”**

**Disciplina: “Structuri de date și algoritmi”**

Varianta 4

**A efectuat:**

Student grupa TI-231 FR

Apareci Aurica

**A verificat:**

Asistent universitar

Mantaluță Marius

**Chișinău 2024**

## Cuprins

1.	Cadrul teoretic .....	3
2.	Listingul programului .....	3
3.	Testarea aplicației .....	8
4.	Concluzii .....	10

## 1. Cadrul teoretic

**Scopul:** Programarea algoritmilor de prelucrare a fișierelor textuale prin utilizarea structurilor, funcțiilor, pointerilor, alocării dinamice a memoriei în limbajul C.

**Sarcina:** Pentru fișierele textuale cu elemente de tip structură (conform variantelor) să se afișeze la ecran următorul meniu de opțiuni:

1. Crearea unui fișier textual nou, introducerea câmpurilor structurilor de la tastatură și înscriserea lor în acest fișier.
2. Afișarea elementelor fișierului binar la ecran.
3. Adăugarea unei structuri noi la sfârșit de fișier.
4. Modificarea unei structuri a fișierului.
5. Căutarea după un câmp al structurii în fișier.
6. Sortarea structurilor fișierului după un careva câmp de tip string & integer(float).
7. Eliminarea unei structuri din fișier.
0. Ieșire din program.

Să se elaboreze funcțiile pentru implementarea opțiunilor meniului.

**Varianta 4.** Structura *Automobil* cu câmpurile: modelul, țara, data fabricării, puterea motorului, costul.

## 2. Listingul programului

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

main.c

```
int main()
{
    while (go)
    {
        userChose = Menu();
        BL();
    }
    return 0;
}
```

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

struct.h

```
typedef struct Car {
    char model[50];
    char country[30];
```

```

char manufacturing_date[15];
float engine_power;
float cost;
} Car;

Car * ReadData()
{
    Car *c = (Car *)malloc(sizeof(Car));
    printf("Introduceti modelul: ");
    fflush(stdin);
    gets(c->model);
    printf("Introduceti tara de provenienta: ");
    fflush(stdin);
    gets(c->country);
    printf("Introduceti data de fabricatie: ");
    fflush(stdin);
    gets(c->manufacturing_date);
    printf("Introduceti puterea motorului: ");
    fflush(stdin);
    scanf("%f", &c->engine_power);
    printf("Introduceti costul: ");
    fflush(stdin);
    scanf("%f", &c->cost);
    return c;
}

void PrintData(Car *c)
{
    printf("\n-----\n");
    printf("Modelul: %s\n", c->model);
    printf("Tara de provenienta: %s\n", c->country);
    printf("Data de fabricatie: %s\n", c->manufacturing_date);
    printf("Puterea motorului: %.2f\n", c->engine_power);
    printf("Costul: %.2f\n", c->cost);
    printf("\n-----\n");
}

void UpdateData(Car * c)
{
    char resp;
    printf("Modificati modelul?(y/n) ->");
    fflush(stdin);
    scanf("%c", &resp);
    if (tolower(resp) == 'y')
    {
        fflush(stdin);
        printf("Modelul nou:\t");
        scanf("%s", c->model);
    }
    printf("Modificati tara de provenienta?(y/n) ->");
    fflush(stdin);
    scanf("%c", &resp);
    if (tolower(resp) == 'y')
    {
        fflush(stdin);

```

```

        printf("Tara de provenienta noua:\t");
        scanf("%s", c->country);
    }
    printf("Modificati data de fabricare?(y/n) ->");
    fflush(stdin);
    scanf("%c", &resp);
    if (tolower(resp) == 'y')
    {
        fflush(stdin);
        printf("Data de fabricare noua:\t");
        scanf("%s", c->manufacturing_date);
    }
    printf("Modificati puterea motorului?(y/n) ->");
    fflush(stdin);
    scanf("%c", &resp);
    if (tolower(resp) == 'y')
    {
        fflush(stdin);
        printf("Puterea noua a motorului:\t");
        scanf("%d", &c->engine_power);
    }
    printf("Modificati costul?(y/n) ->");
    fflush(stdin);
    scanf("%c", &resp);
    if (tolower(resp) == 'y')
    {
        fflush(stdin);
        printf("Costul nou:\t");
        scanf("%f", &c->cost);
    }
}

Car * FromString(char * str)
{
    Car * c = (Car *)malloc(sizeof(Car));
    sscanf(str, "%[^_]_|_%[^_]_|_%[^_]_|_%f_|_%f\n", c->model, c->country, c->manufacturing_date, &c->engine_power, &c->cost);
    return c;
}

char * ToString(Car c)
{
    char * str = (char *)malloc(103);
    sprintf(str, "%s_|_%s_|_%s_|_%f_|_%f\n", c.model, c.country, c.manufacturing_date, c.engine_power, c.cost);
    return str;
}

void swap(Car *a, Car *b)
{
    Car aux = *a;
    *a = *b;
    *b = aux;
}

```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include <conio.h>
#include "struct.h"

Car * _masini;
int nr = 0;

void Read(){
    printf("Dati numarul de masini ce urmeaza a fi introduse: ");
    scanf("%d", &nr);
    _masini = (Car *)malloc(nr * sizeof(Car));
    for (int i = 0; i < nr; i++)
    {
        printf("Car %d\n-----\n", i + 1);
        Car * c = ReadData();
        _masini[i] = *c;
    }
}

void Show(){
    for (int i = 0; i < nr; i++)
    {
        PrintData(&_masini[i]);
    }
}

void Search(){
    char mod[50];
    printf("Modelul masinii cautate: ");
    scanf("%s", mod);
    for (int i = 0; i < nr; i++)
    {
        if (strcmp(_masini[i].model, mod) == 0)
        {
            PrintData(&_masini[i]);
            return;
        }
    }
    printf("Masina nu a fost gasit\n");
}

void Update(){
    int index = 0;
    printf("Introduceti indexul masinii ce urmeaza a fi modificata: ");
    scanf("%d", &index);
    if (index < 0 || index >= nr)
    {
        printf("Indexul introdus nu este valid\n");
        return;
    }
    UpdateData(&_masini[index]);
}
```

```

void SortList(){
    for (int i = 0; i < nr - 1; i++)
    {
        for (int j = i + 1; j < nr; j++)
        {
            if (_masini[i].engine_power > _masini[j].engine_power)
            {
                swap(&_masini[i], &_masini[j]);
            }
        }
    }
    printf("Masinile au fost sortate dupa puterea motorului\n");
}

```

```

void Delete(){
    int index = 0;
    printf("Introduceti indexul masinii ce urmeaza a fi sterse: ");
    scanf("%d", &index);
    if (index < 0 || index >= nr)
    {
        printf("Indexul introdus nu este valid\n");
        return;
    }
    for (int i = index; i < nr - 1; i++)
    {
        _masini[i] = _masini[i + 1];
    }
    nr--;
    _masini = (Car *)realloc(_masini, nr * sizeof(Car));
}

```

```

void New(){
    nr++;
    _masini = (Car *)realloc(_masini, nr * sizeof(Car));
    printf("Car %d\n\n-----\n", nr);
    Car * c = ReadData();
    _masini[nr - 1] = *c;
}

```

```

void Import(){
    FILE * f = fopen("Cars.txt", "r");
    if (f == NULL)
    {
        printf("Fisierul nu a fost gasit\n");
        return;
    }
    fread(&nr, sizeof(int), 1, f);
    _masini = (Car *)malloc(nr * sizeof(Car));
    char * buffer = (char *)malloc(103);
    for(int i = 0; i < nr; i++)
    {
        fgets(buffer, 103, f);
        _masini[i] = *FromString(buffer);
    }
    fclose(f);}

```

### 3. Testarea aplicației

```

1 .   Citire date masina
2 .   Afisare date masina
3 .   Cautare masina
4 .   Actualizare date masina
5 .   Sortare masini
6 .   Stergere masina
7 .   Adauga masina la sfarsit
8 .   Import masini din fisier text
9 .   Export masini din fisier text
10 .  Sterge fisierul text
11 .  Elibereaza memoria
0 .   Exit
-----

```

Optiunea aleasa -->

Nr.	Input	Output
1.	Introducere date despre mașini	<p>Dati numarul de masini ce urmeaza a fi introduse: 2 Car 1</p> <p>-----</p> <p>Introduceti modelul: Toyota Supra MK4 Introduceti tara de provenienta: Japonia Introduceti data de fabricatie: 13/11/1997 Introduceti puterea motorului: 1200 Introduceti costul: 60000 Car 2</p> <p>-----</p> <p>Introduceti modelul: Nisan Silvai S15 Introduceti tara de provenienta: Japonia Introduceti data de fabricatie: 16/05/1995 Introduceti puterea motorului: 980 Introduceti costul: 45000</p>
2.	Afișare date despre mașini	<p>-----</p> <p>Modelul: Toyota Supra Mk4 Tara de provenienta: Japonia Data de fabricatie: 13/11/1997 Puterea motorului: 1200.00 Costul: 60000.00</p> <p>-----</p> <p>-----</p> <p>Modelul: Nisan Silvia S15 Tara de provenienta: Japonia Data de fabricatie: 16/05/1995 Puterea motorului: 980.00 Costul: 45000.00</p> <p>-----</p>



3.	Căutarea unui element	<pre> Modelul masinii cautate: x Masina nu a fost gasit  Modelul masinii cautate: 1  -----  Modelul: 1 Tara de provenienta: 1 Data de fabricatie: 1 Puterea motorului: 1.00 Costul: 1.00  ----- </pre>	
4.	Actualizare date despre mașină	<pre> Introduceti indexul masinii ce urmeaza a fi modificata: 1 Modificati modelul?(y/n) -&gt;n Modificati tara de provenienta?(y/n) -&gt;n Modificati data de fabricare?(y/n) -&gt;n Modificati puterea motorului?(y/n) -&gt;y Puterea noua a motorului:      1000 Modificati costul?(y/n) -&gt;y Costul nou:      45500 </pre>	
5.	Sortare obiecte.	<pre> Masinile au fost sortate dupa puterea motorului  -----  Modelul: Nisan Silvia S15 Tara de provenienta: Japonia Data de fabricatie: 16/05/1995 Puterea motorului: 980.00 Costul: 45000.00  -----  Modelul: Toyota Supra Mk4 Tara de provenienta: Japonia Data de fabricatie: 13/11/1997 Puterea motorului: 1200.00 Costul: 60000.00  ----- </pre>	
6.	Ștergerea unui element	<pre> Introduceti indexul masinii ce urmeaza a fi sterse: 2 </pre>	
7.	Adăugare mașină la sfârșit	<pre> Car 3  -----  Introduceti modelul: carToRemove Introduceti tara de provenienta: MD Introduceti data de fabricatie: 12/12/2000 Introduceti puterea motorului: 980 Introduceti costul: 200000 </pre>	
8.	Import/Export date în fișier	<pre> # Cars.txt 1 STANULIUK Toyota Supra Mk4_ _Japonia_ _13/11/1997_ _1200.000000_ _60000.000000 2 Nisan Silvia S15_ _Japonia_ _16/05/1995_ _980.000000_ _45000.000000 3 </pre>	

## 4. Concluzii

În cadrul acestei lucrări de laborator, am explorat și implementat manipularea fișierelor binare în limbajul de programare C. Această abordare ne-a oferit o modalitate eficientă de salvare și recuperare a datelor într-un format compact, potrivit pentru structuri de date complexe. Un exemplu concret a fost reprezentat de structura definită pentru informațiile despre automobile, cuprinzând detalii precum modelul, țara de proveniență, data fabricării, puterea motorului și costul.

Unul dintre obiectivele principale ale programului dezvoltat a fost gestionarea unor operații fundamentale asupra fișierelor binare. Astfel, am implementat funcționalități precum crearea unui fișier binar, afișarea conținutului, adăugarea de noi înregistrări, modificarea structurilor existente, căutarea după anumite câmpuri și eliminarea unei structuri din fișier. În ceea ce privește gestionarea erorilor, am inclus verificări esențiale pentru a asigura corectitudinea și stabilitatea programului. Aceasta implică, printre altele, verificarea dacă un fișier poate fi deschis cu succes sau nu.

În concluzie, această lucrare de laborator ne-a furnizat o bază solidă pentru manipularea fișierelor binare în limbajul C, evidențiind, totodată, posibilități de îmbunătățire. Astfel, există perspective de dezvoltare ulterioară, precum adăugarea de funcționalități suplimentare, gestionarea excepțiilor mai detaliate sau implementarea altor operații complexe pentru a extinde capacitățile programului în gestionarea datelor binare.