

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



# Raport

**la lucrarea de laborator nr. 3**

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

**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 .....	10
4.	Concluzii .....	12

## 1. Cadrul teoretic

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

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

1. Crearea unui fișier binar 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.

7. Eliminarea unei structuri din fișier.

8. Ștergerea fișierului de pe disc (la dorință)

0. Ieșire din program.

Să se elaboreze funcțiile pentru implementarea opțiunilor meniului.

**Varianta 4.** Structura *Cinematograf* cu câmpurile: denumirea, adresa, telefonul, numărul de locuri, prețul biletelor.

## 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 Cinematograf {
    char denumirea[50];
    char adresa[100];
```

```

    char telefonul[15];
    int numarLocuri;
    float pretBilete;
} Cinematograf;

Cinematograf * ReadData(){
    Cinematograf *c = (Cinematograf *)malloc(sizeof(Cinematograf));
    printf("Denumirea: ");
    fflush(stdin);
    scanf("%s", c->denumirea);
    printf("Adresa: ");
    fflush(stdin);
    scanf("%s", c->adresa);
    printf("Telefonul: ");
    fflush(stdin);
    scanf("%s", c->telefonul);
    printf("Numarul de locuri: ");
    fflush(stdin);
    scanf("%d", &c->numarLocuri);
    printf("Pretul biletelor: ");
    fflush(stdin);
    scanf("%f", &c->pretBilete);
    return c;
}

void PrintData(Cinematograf *c){
    printf("\n-----\n");
    printf("Denumirea: %s\n", c->denumirea);
    printf("Adresa: %s\n", c->adresa);
    printf("Telefonul: %s\n", c->telefonul);
    printf("Numarul de locuri: %d\n", c->numarLocuri);
    printf("Pretul biletelor: %.2f\n", c->pretBilete);
    printf("\n-----\n");
}

void UpdateData(Cinematograf * c){
    char resp;
    printf("Modificati denumirea?(y/n) ->");
    fflush(stdin);
    scanf("%c", &resp);
    if (tolower(resp) == 'y')
    {
        fflush(stdin);
        printf("Denumirea noua:\t");
        scanf("%s", c->denumirea);
    }

    printf("Modificati adresa?(y/n) ->");
    fflush(stdin);
    scanf("%c", &resp);
    if (tolower(resp) == 'y')
    {
        fflush(stdin);
        printf("Adresa noua:\t");
        scanf("%s", c->adresa);
    }
}

```

```

    }

    printf("Modificati telefonul?(y/n) ->");
    fflush(stdin);
    scanf("%c", &resp);
    if (tolower(resp) == 'y')
    {
        fflush(stdin);
        printf("Telefonul nou:\t");
        scanf("%s", c->telefonul);
    }

    printf("Modificati numarul de locuri?(y/n) ->");
    fflush(stdin);
    scanf("%c", &resp);
    if (tolower(resp) == 'y')
    {
        fflush(stdin);
        printf("Numarul de locuri nou:\t");
        scanf("%d", &c->numarLocuri);
    }

    printf("Modificati pretul biletelor?(y/n) ->");
    fflush(stdin);
    scanf("%c", &resp);
    if (tolower(resp) == 'y')
    {
        fflush(stdin);
        printf("Pretul biletelor nou:\t");
        scanf("%f", &c->pretBilete);
    }
}

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

```

```

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

```

```

Cinematograf * _cinematografe;
int nr = 0;

```

```

void Read(){
    printf("Dati numarul de cinematografe ce urmeaza a fi introduse: ");
    scanf("%d", &nr);
    _cinematografe = (Cinematograf *)malloc(nr * sizeof(Cinematograf));
    for (int i = 0; i < nr; i++)

```

user.h

```

    {
        printf("Cinematograf %d\n-----\n", i + 1);
        Cinematograf * c = ReadData();
        _cinematografe[i] = *c;
    }
}

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

void Search(){
    char denumirea[50];
    printf("Denumirea cinematografului cautat: ");
    scanf("%s", denumirea);
    for (int i = 0; i < nr; i++)
    {
        if (strcmp(_cinematografe[i].denumirea, denumirea) == 0)
        {
            PrintData(&_cinematografe[i]);
            return;
        }
    }
    printf("Cinematograful nu a fost gasit\n");
}

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

void SortList(){
    for (int i = 0; i < nr - 1; i++)
    {
        for (int j = i + 1; j < nr; j++)
        {
            if (_cinematografe[i].numarLocuri > _cinematografe[j].numarLocuri)
            {
                swap(&_cinematografe[i], &_cinematografe[j]);
            }
        }
    }
    printf("Cintematografele au fost sortate descrescator dupa numarul de locuri\n");
}

```

```

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

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

void Import(){
    FILE * f = fopen("cinematografe.bin", "rb");
    if (f == NULL)
    {
        printf("Fisierul nu a fost gasit\n");
        return;
    }
    fread(&nr, sizeof(int), 1, f);
    _cinematografe = (Cinematograf *)malloc(nr * sizeof(Cinematograf));
    fread(_cinematografe, sizeof(Cinematograf), nr, f);
    fclose(f);
}

void Export(){
    FILE * f = fopen("cinematografe.bin", "wb");
    fwrite(&nr, sizeof(int), 1, f);
    fwrite(_cinematografe, sizeof(Cinematograf), nr, f);
    fclose(f);
}

void FileDelete(){
    if (remove("cinematografe.bin") == 0)
    {
        printf("Fisierul a fost sters cu succes\n");
    }
    else
    {
        printf("Fisierul nu a putut fi sters\n");
    }
}

```

```

    }

}

int userChose = 0;
int go = 1;

int Menu(){
    printf("-----Meniul Aplicatiei-----\n");
    printf("1 . \tCitire cinematografe\n");
    printf("2 . \tAfisare cinematografe\n");
    printf("3 . \tCautare cinematograf\n");
    printf("4 . \tActualizare cinematograf\n");
    printf("5 . \tSortare cinematografe\n");
    printf("6 . \tStergere cinematograf\n");
    printf("7 . \tAdauga cinematograf la sfarsit\n");
    printf("8 . \tImport cinematografe din fisier binar\n");
    printf("9 . \tExport cinematografe din fisier binar\n");
    printf("10 . \tSterge fisierul binar\n");
    printf("11 . \tElibereaza memoria\n");
    printf("0 . \tExit\n");
    printf("-----\n");
    printf("\nOptiunea aleasa --> ");
    int op;
    scanf("%d", &op);
    system("cls");
    return op;
}

void PressAnyKey(){
    printf("\nAtingeti o tasta pentru a continua\n");
    getch();
    system("cls");
}

void BL(){
    switch (userChose)
    {
        case 1:
        {
            Read();
            PressAnyKey();
        } break;
        case 2:
        {
            Show();
            PressAnyKey();
        }break;
        case 3:
        {
            Search();
            PressAnyKey();
        }break;
        case 4:
        {

```



```

        Update();
        PressAnyKey();
    }break;
    case 5:
    {
        SortList();
        PressAnyKey();
    }break;
    case 6:
    {
        Delete();
        PressAnyKey();
    }break;
    case 7:
    {
        New();
        PressAnyKey();
    }break;
    case 8:
    {
        Import();
        PressAnyKey();
    }break;
    case 9:
    {
        Export();
        PressAnyKey();
    }break;
    case 10:
    {
        FileDelete();
        PressAnyKey();
    }break;
    case 11:
    {
        free(_cinematografe);
        nr = 0;
        PressAnyKey();
    }break;
    case 0:
    {
        go=0;
        system("cls");
        printf("Aplicatia s-a oprit cu succes");
        getch();
    }break;
    default:printf("Optiune necunoscuta\nIncercati din nou");break;
}
}

```

### 3. Testarea aplicației

```
-----Meniul Aplicatiei-----
1 .   Citire cinematografe
2 .   Afisare cinematografe
3 .   Cautare cinematograf
4 .   Actualizare cinematograf
5 .   Sortare cinematografe
6 .   Stergere cinematograf
7 .   Adauga cinematograf la sfarsit
8 .   Import cinematografe din fisier binar
9 .   Export cinematografe din fisier binar
10 .  Sterge fisierul binar
11 .  Elibereaza memoria
0 .   Exit
-----
Optiunea aleasa -->
```

Nr.	Input	Output
1.	Introducerea datelor	Dati numarul de cinematografe ce urmeaza a fi introduse: 3 Cinematograf 1 ----- Denumirea: Cineplex-Loteanu Adresa: bd_Stefan_cel_Mare Telefonul: 0123456789 Numarul de locuri: 100 Pretul biletelor: 80 Cinematograf 2 ----- Denumirea: AutoCinema Adresa: str_Mihail_Sadoveanu Telefonul: 0123456789 Numarul de locuri: 120 Pretul biletelor: 150
2.	Afișarea informațiilor	Alegeti o optiune: 2 ===== Cinematograf 1: Denumirea: Cineplex-Loteanu Adresa: bd_Stefan_cel_Mare Telefonul: 012345678 Numar locuri: 100 Pret bilete: 80.00 ===== Cinematograf 2: Denumirea: AutoCinema Adresa: str_Mihai_Sadoveanu Telefonul: 012345678 Numar locuri: 120 Pret bilete: 150.00 ===== Cinematograf 3: Denumirea: Cinematograful_Luceafarul Adresa: Nisporeni Telefonul: 012345678 Numar locuri: 80 Pret bilete: 100.00

3.	Căutarea unui element	Denumirea cinematografului cautat: AutoCinema  ----- Denumirea: AutoCinema Adresa: str_Mihail_Sadoveanu Telefonul: 0123456789 Numarul de locuri: 120 Pretul biletelor: 150.00  ----- Denumirea cinematografului cautat: x Cinematograful nu a fost gasit
4.	Actualizarea unui element	Introduceti indexul cinematografului ce urmeaza a fi modificat: 1 Modificati denumirea?(y/n) ->y Denumirea noua: cinematograf_nou Modificati adresa?(y/n) ->n Modificati telefonul?(y/n) ->n Modificati numarul de locuri?(y/n) ->n Modificati pretul biletelor?(y/n) ->y Pretul biletelor nou: 330  ----- Denumirea: cinematograf_nou Adresa: str_Mihail_Sadoveanu Telefonul: 0123456789 Numarul de locuri: 120 Pretul biletelor: 330.00
5.	Sortarea cinematografelor	Cintematografele au fost sortate descrescator dupa numarul de locuri
6.	Ștergerea unui cinematograf	Introduceti indexul cinematografului ce urmeaza a fi sters: 2
7.	Adăugarea unui cinematograf	Cinematograf 3  ----- Denumirea: Cinematograf_3 Adresa: 3 Telefonul: 3 Numarul de locuri: 3 Pretul biletelor: 3
8.	Import/Export date în fișier binar	❏ cinematografe.bin X ❏ cinematografe.bin 1 nulstr STORLESTSTRTKRLSTYBLRLESTBRLESTTNULSTORLESTXSDMETRLESTXSOMETX STXSOMETX STX STORLESTBRLRLESTKRLASTYBLRLESTBRLESTTNULSTNLASTXTSTKTRLESTV

## 4. Concluzii

În cadrul acestei lucrări de laborator, am abordat subiectul lucrului cu fișierele binare în limbajul de programare C, în contextul unui program de gestionare a datelor despre cinematografe. Prin intermediul acestui proiect, am implementat funcționalități precum crearea unui fișier binar, afișarea conținutului acestuia, adăugarea unei noi structuri la sfârșitul fișierului, modificarea unei structuri existente, căutarea după un anumit câmp, sortarea structurilor după un câmp specific, eliminarea unei structuri și ștergerea completă a fișierului.

Am început prin a defini o structură pentru datele cinematografelor și am utilizat un array global pentru a stoca aceste structuri, facilitând astfel manipularea datelor în mai multe funcții. Am implementat funcții care deschid și închid fișierele binare, precum și funcții pentru citirea și scrierea datelor în/din aceste fișiere. De asemenea, am integrat și un algoritm de sortare rapidă (BubbleSort) pentru a evidenția capacitatea programului de a organiza datele cinematografelor în funcție de numărul de locuri disponibile.

În concluzie, lucrul cu fișierele binare în limbajul C necesită manipulări atente ale fișierelor, utilizând funcții adecvate pentru deschiderea, citirea, scrierea și închiderea acestora. Proiectul dezvoltat oferă o serie de funcționalități utile pentru gestionarea datelor despre cinematografe, evidențiind practicile de programare eficiente și organizarea structurată a codului.