Министерство образования Республики Беларусь

г. Минск

Государственное учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

**Лабораторная работа №4**

**“ Объединения.”**

**Учебная группа 230501**

Выполнил: Кочеров Роман Сергеевич

Проверил: Селезнев Александр Игоревич

2023 год

**Вариант 4**

Структура содержит информацию о фильмах: длительность в минутах (число), название (указатель), вложенное объединение – жанр (строка фиксированной длины) и год выпуска. Найти *самые новые фильмы*. Удалить фильмы *заданного жанра*.

1. **Организовать структуру, содержащую объединение** (по варианту), в *main* объявить **динамический массив структур**, проверить выделение памяти и освободить её в конце программы.
2. Дополнить **функции для ввода и вывода** первых двухполей структуры из ЛР1 вводом и выводом **остальных полей**.
3. Написать **функцию поиска** в массиве структур **по заданному**.
4. Написать **функцию удаления** структур из массива **по заданному**.
5. Для решения задачи организовать **меню для работы с массивом структур** и всевозможные **проверки на корректность** получаемых данных.

#include "Roma.h";

int main() {

int num\_movies=0, choice=0;

int \*arr\_data = (int\*)malloc(100 \* sizeof(int));

Movie\* movies = NULL;

Movie\* new\_films;

printf("\n-------------------------\n");

while (choice != 5)

{

printf("enter film(1) \ndisplay films(2) \nfind new film(3) \ndelete films(4) \nexit(5)\n\n ");

choice = vvod();

switch (choice)

{

case 1:

{

new\_films = (struct movie\*)realloc(movies, (num\_movies + 1) \* sizeof(Movie));

if (new\_films == NULL) {

printf("Error: Memory allocation failed\n");

break;}

movies = new\_films;

input\_film\_info(&movies[num\_movies], arr\_data, num\_movies);

num\_movies++;

printf("Film added successfully!\n");

break;

}

break;

case 2:

print\_movies\_info(movies, num\_movies, arr\_data);

break;

case 3:

find\_new\_movies(movies, num\_movies, arr\_data);

break;

case 4:

remove\_movies(movies, &num\_movies);

break;

case 5:

goodbay(movies, num\_movies, arr\_data);

break;

}

printf("\n-------------------------\n");

}

return 0;

}

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX\_TITLE\_LENGTH 100

#define MAX\_GENRE\_LENGTH 20

typedef struct {

int duration;

char\* title;

union

{

int release\_year;

char genre[MAX\_GENRE\_LENGTH];

} info;

} Movie;

void print\_movies\_info(Movie\* movies, int num\_movies, int arr[])

{

for (int i = 0; i < num\_movies; i++)

{

printf("Movie #%d:\n", i + 1);

printf(" Title: %s\n", movies[i].title);

printf(" Duration: %d minutes\n", movies[i].duration);

if (strlen(movies[i].info.genre) > 0) {

printf(" Genre: %s\n", movies[i].info.genre);

}

else {

printf(" Genre: N/A\n");

}

printf(" Release year: %d\n", arr[i]);

}

return 0;

}

void input\_film\_info(Movie\* film, int\* arr, int x)

{

printf("Please enter the duration (in minutes) of the film: ");

film->duration = vvod();

film->title = malloc(MAX\_TITLE\_LENGTH \* sizeof(char));

printf("Please enter the title of the film: ");

fgets(film->title, MAX\_TITLE\_LENGTH, stdin);

film->title[strcspn(film->title, "\n")] = '\0'; // remove newline character from input

printf("Enter the release year of movie: ");

film->info.release\_year = vvod();

arr[x] = film->info.release\_year;

printf("Enter the genre of movie: ");

char genre[MAX\_GENRE\_LENGTH];

scanf("%s", genre);

strcpy(film->info.genre, genre);

return 0;

}

void find\_new\_movies(Movie\* movies, int num\_movies, int arr\_data[])

{

printf("Newest movies:\n");

int newest\_year = 0;

for (int i = 0; i < num\_movies; i++) {

if (arr\_data[i] > newest\_year) {

newest\_year = arr\_data[i];

}

}

for (int i = 0; i < num\_movies; i++) {

if (arr\_data[i] == newest\_year) {

printf("- %s\n", movies[i].title);

}

}

return 0;

}

int vvod()

{

int var;

while ((scanf\_s("%d", &var) == 0) || getchar() != '\n')

{

printf("wrong input! try again\n");

rewind(stdin);

}

return var;

}

void remove\_movies(Movie movies[], int\* num\_movies)

{

printf("Enter the genre of movies to remove: ");

char genre\_to\_remove[MAX\_GENRE\_LENGTH];

scanf("%s", genre\_to\_remove);

int i = 0;

while (i < \*num\_movies) { // пока не пройдемся по всем фильмам

if (strcmp(movies[i].info.genre, genre\_to\_remove) == 0) { // если продолжительность больше максимальной

for (int j = i; j < \*num\_movies - 1; j++) { // сдвигаем все элементы массива на один влево

movies[j] = movies[j + 1];

}

(\*num\_movies)--; // уменьшаем количество фильмов на 1

}

else {

i++; // переходим к следующему фильму

}

}

return 0;

}

void goodbay(Movie\* movies, int num\_movies, int arr\_data[])

{

printf("\n goodbay");

// free memory

for (int i = 0; i < num\_movies; i++) {

free(movies[i].title);

}

free(movies);

free(arr\_data);

}

