

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра Вычислительной техники

ОТЧЕТ
по лабораторной работе № 7
по дисциплине «Программирование»
ТЕМА: Указатели на структуры и функции

Студент гр. 3311

Баймухамедов Р.Р

Преподаватель

Хахаев И.А.

Санкт-Петербург

2024

Цель работы

Целью работы является изучение и освоение работы с указателями на структуры и на функции.

Задание (вариант 8)

Для выбранной предметной области создать динамический массив структур, содержащих характеристики объектов предметной области.

Обязательный набор полей:

- динамический массив символов, включая пробелы (name)
- произвольный динамический массив символов
- числовые поля типов `int` и `float` (не менее двух полей каждого типа)
- поле с числовым массивом.

Написать программу, обеспечивающую начальное формирование массива структур при чтении из файла (текст с разделителями — CSV) с последующим возможным дополнением элементов массива при вводе с клавиатуры. Следует использовать указатели на структуры и указатели на функции обработки массива в соответствии с вариантом задания.

Во всех случаях, когда при поиске записей результат отсутствует, следует вывести сообщение.

Выбор записей, в которых значение любого символьного поля (выбор из меню) начинается с указанной подстроки без учета регистра, сортировка результата по убыванию значений последнего числового поля.

Постановка задачи и описание решения

Для выполнения данной лабораторной работы необходимо разработать программу, которая будет читать данные из файла и записывать их в массив ARR структур типа MOV и упорядочивать их по убыванию значений последнего числового поля, затем запрашивать у пользователя подстроку и символьное поле, в котором будет производится поиск на наличие подстроки, и при вхождении подстроки в строку выбранного поля выводить данные массива. Предметная область будет фильмы: название, режиссёр, год выхода, длительность, рейтинг КиноПоиска, личный рейтинг и дата просмотра

Для достижения этой цели необходимо выполнить следующие шаги:

1. Запросить у пользователя символ-разделитель
2. Посчитать количество строк в файле

3. Прочитывать данные строки файла, разделенные введенным ранее символом-разделителем, записывать их во временный массив TMP, затем данные строки записывать в массив ARR структуры MOV, до тех пор, пока данные всех строк не будут записаны в массив ARR
4. Упорядочить записи массива ARR по убыванию значений последнего числового поля (дате дня просмотра)
5. Запросить у пользователя по какому полю структуры будет производится поиск
6. Запросить у пользователя подстроку
7. Вывести, при вхождении подстроки в строку по выбранному полю структуры, информацию о фильме (данные массива)

Описание функций

№	Функция	Тип	Назначение
1	main	int	Запускает программу, обрабатывает пользовательский ввод, вызывает другие функции в зависимости от выбора пользователя, и завершает программу при необходимости.
2	menu	void	Выводит варианты выбора пользователя
3	print_head	void	Вывод оглавления таблицы данных
4	output_array	void	Вывод массива
5	count_of_lines	int	Подсчёт количества строк в файле
6	split_string	void	Разделяет строку по символу-разделителю и записывает элементы строки в массив
7	add_to_array	void	Чтение из файла и запись элементов строк в предназначенный массив, имеющий структуру MOV
8	new_gets	void	Посимвольный ввод строки
9	add_new_data	void	Добавление нового элемента структуры MOV в файл
10	sort_array	void	Сортировка массива, имеющего структуру MOV, по убыванию значений последнего числового поля (по дате просмотра)
11	find_name	char	Возвращает название фильма в массиве под указанным индексом
12	find_director	char	Возвращает режиссёра фильма в массиве под указанным индексом
13	find_by	void	Ввод подстроки и проверка строк выбранного поля структуры на её наличие, вывод информации массива при успешной проверке
14	clear_stream	void	Очистка входного буфера

Описание переменных

int main()

№	Переменная	Тип	Назначение
1	**ARR	MOV (struct movie)	Массив, структуры типа MOV, в котором будут храниться данные
2	i	int	Кол-во строк в файле
3	option	int	Опция, выбираемая пользователем
4	sep	char	Символ-разделитель

struct movie

№	Название	Тип	Назначение
1	*name	char	Название фильма
2	*director	char	Имя и фамилия режиссёра фильма
3	year		Год выхода фильма
4	duration		Длительность фильма
5	kpr		Оценка фильма на КиноПоиске
6	plr		Личная оценка фильма
7	date[3]	int	Дата просмотра фильма

void output_array(MOV *ARR[], int n)

№	Переменная	Тип	Назначение
1	i	int	Переменная в цикле, отвечающая за индекс выводимых данных массива

int count_of_lines(char *filename)

№	Переменная	Тип	Назначение
1	line[maxlen]	char	Хранение информации строки из файла

2	n	int	Счётчик количества строк в файле
---	---	-----	----------------------------------

```
void split_string(char *inputString, char **words, int *wordCount, char delimiter)
```

№	Переменная	Тип	Назначение
1	wordIndex	char	индекс текущего слова в массиве words
2	wordStart	char	индекс начала текущего слова в строке inputString
3	wordLength	char	длина текущего слова
4	inWord	int	флаг, указывающий на то, находится ли функция внутри слова или не внутри
5	i	int	переменная для итерации по символам в строке inputString.

```
void add_to_array(MOV *ARR[], char *filename, int n, char sep)
```

№	Переменная	Тип	Назначение
1	**TMP	char	Временный массив, куда записываются данные из строки файла
2	line[maxlen]	char	Хранение информации строки из файла
3	wordCount	int	Кол-во элементов, на которые строка была разделена
4	i	int	Переменная в цикле, отвечающая за номер элемента массива

```
void new_gets(char s[], int lim)
```

№	Переменная	Тип	Назначение
1	c	char	Вводимый символ
2	i	int	Переменная в цикле, отвечающая за индекс

			вводимого символа в строке
--	--	--	----------------------------

void add_new_data(char *filename)

№	Название	Тип	Назначение
1	name[maxlen]	char	Вводимое название фильма
2	director[maxlen]	char	Вводимые имя и фамилия режиссёра фильма
3	year	int	Вводимый год выхода фильма
4	duration	int	Вводимая длительность фильма
5	kpr	float	Вводимая оценка фильма на КиноПоиске
6	plr	float	Вводимая личная оценка фильма
7	d	int	Вводимый день даты просмотра фильма
8	m	int	Вводимый месяц даты просмотра фильма
9	y	int	Вводимый год даты просмотра фильма

void sort_array(MOV *ARR[], int n)

№	Переменная	Тип	Назначение
1	temp	MOV (struct movie)	Временное хранение данных, структуры типа MOV
2	i	int	Используется для итерации по элементам массива ARR. Индекс текущего элемента, который рассматривается во внешнем цикле.
3	j	int	переменная, используемая для итерации по элементам массива

			ARR во внутреннем цикле. Она представляет собой индекс текущего элемента, с которым сравнивается элемент, находящийся на позиции j+1, для выполнения сортировки по дате рождения.
--	--	--	---

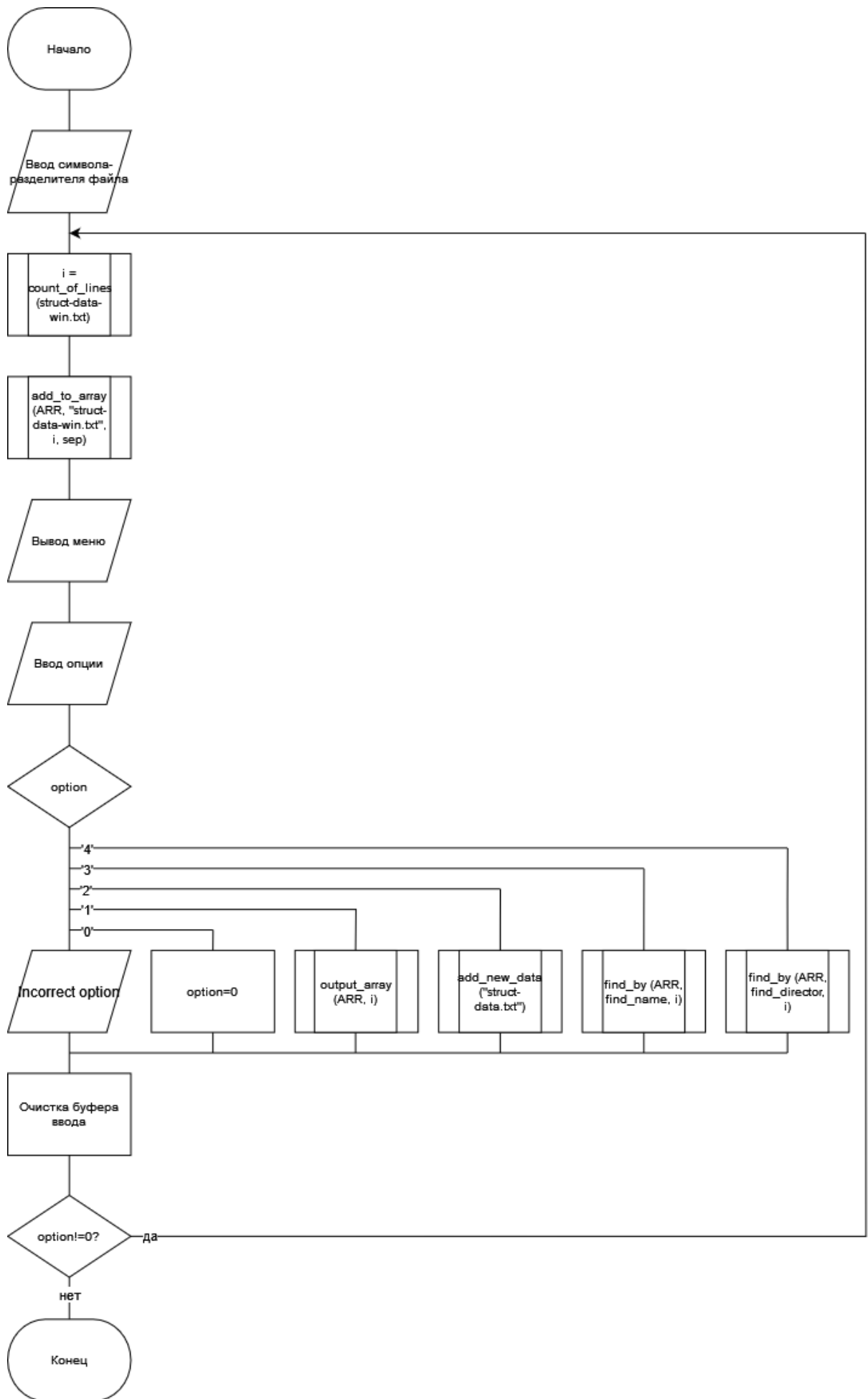
void find_by(MOV *ARR[], char *(*func_name)(MOV**, int), int n)

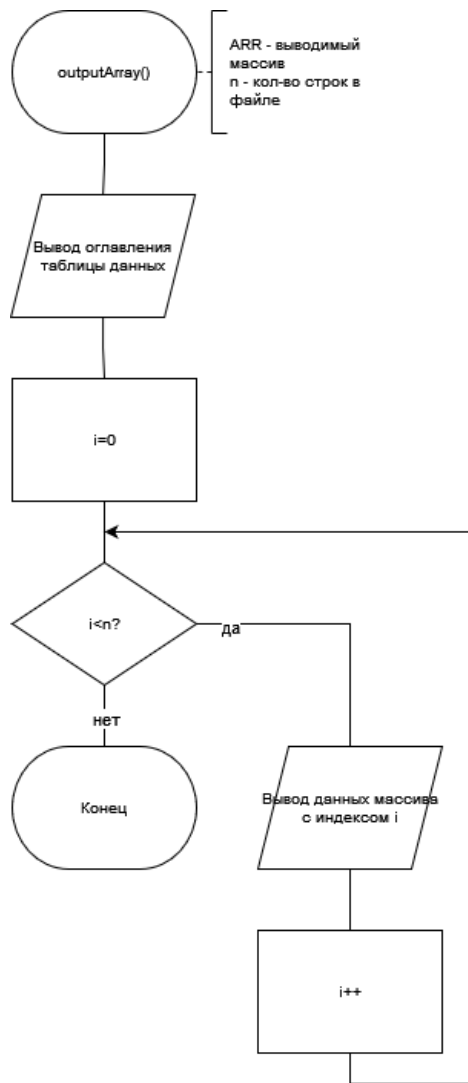
№	Переменная	Тип	Назначение
1	i	int	Переменная в цикле, отвечающая за номер элемента массива
2	result	int	Результат, показывающий входит ли подстрока в строку выбранного поля сравниваемого элемента
3	count	int	Количество элементов, удовлетворяющих условию
4	substring[maxlen]	char	Вводимая подстрока

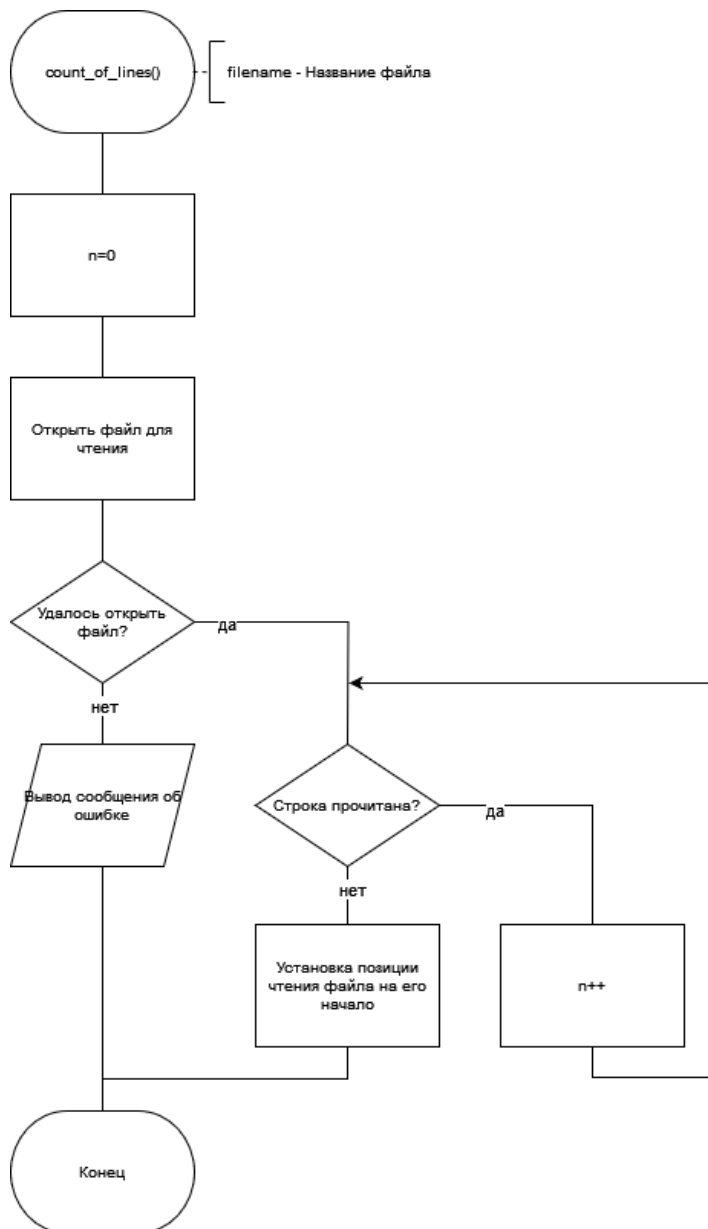
void clear_stream(FILE *input_stream)

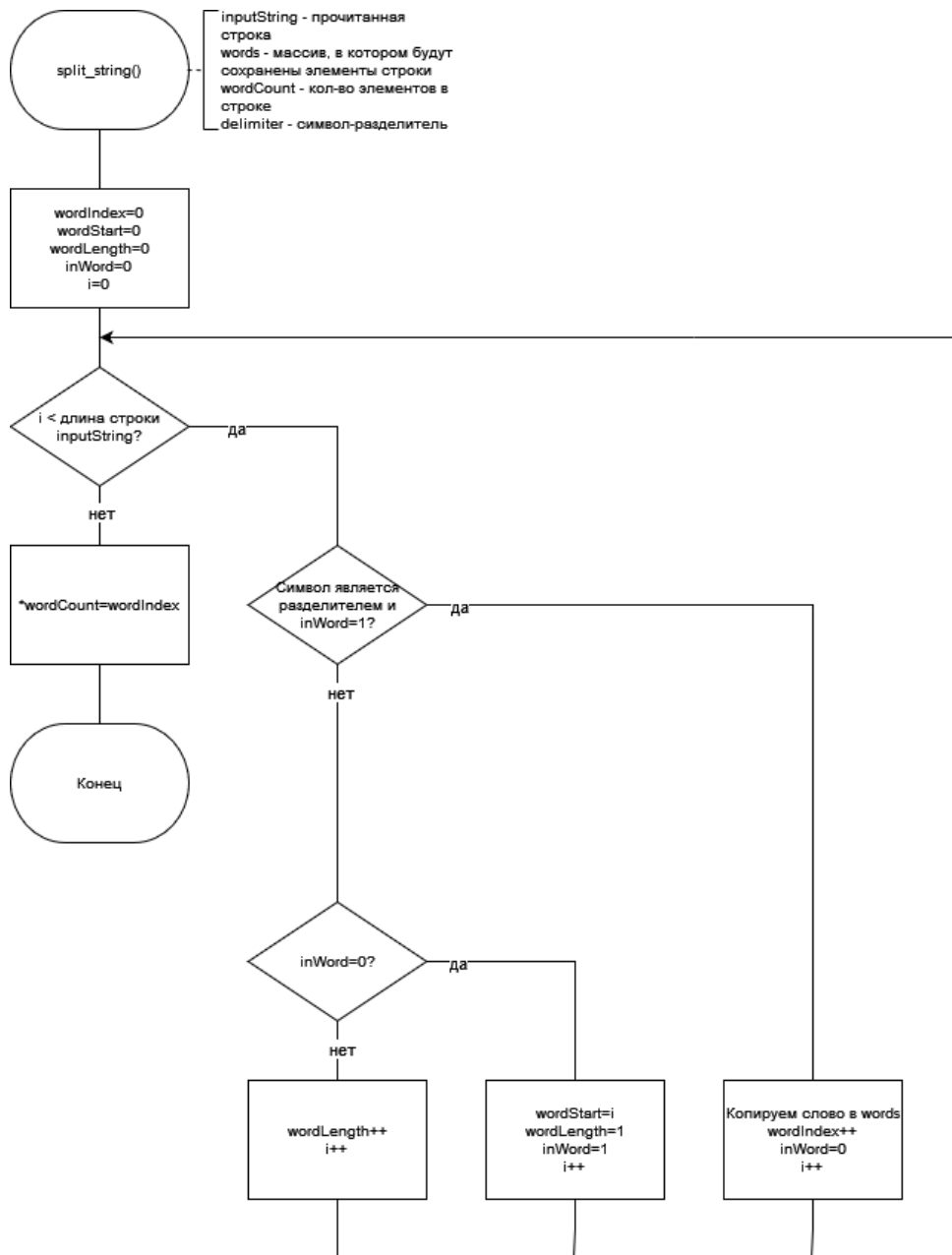
№	Имя переменной	Тип	Назначение
1	input_stream	FILE*	Указатель на поток ввода, который нужно очистить

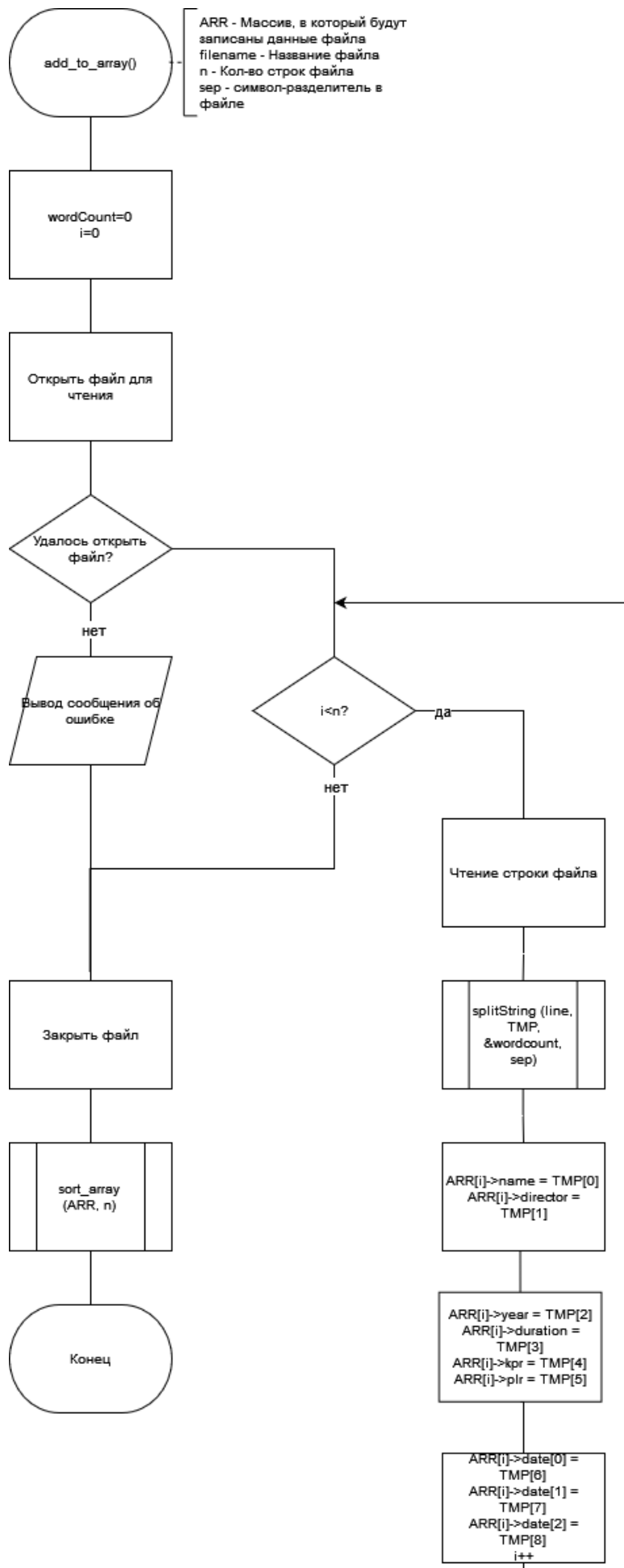
Схемы алгоритмов

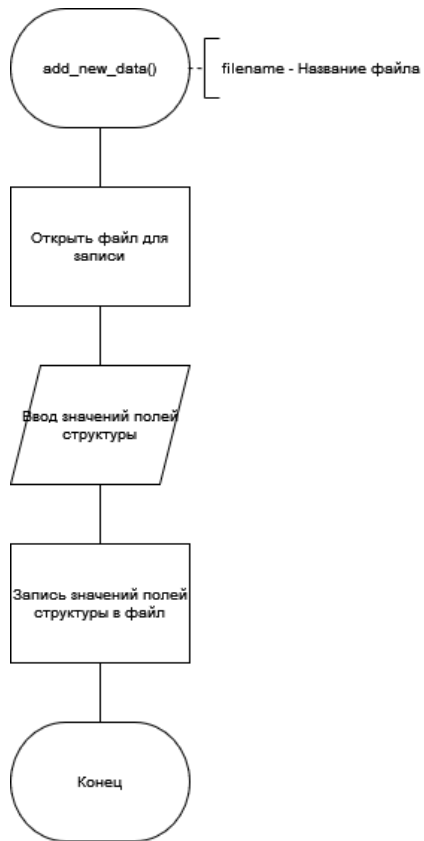


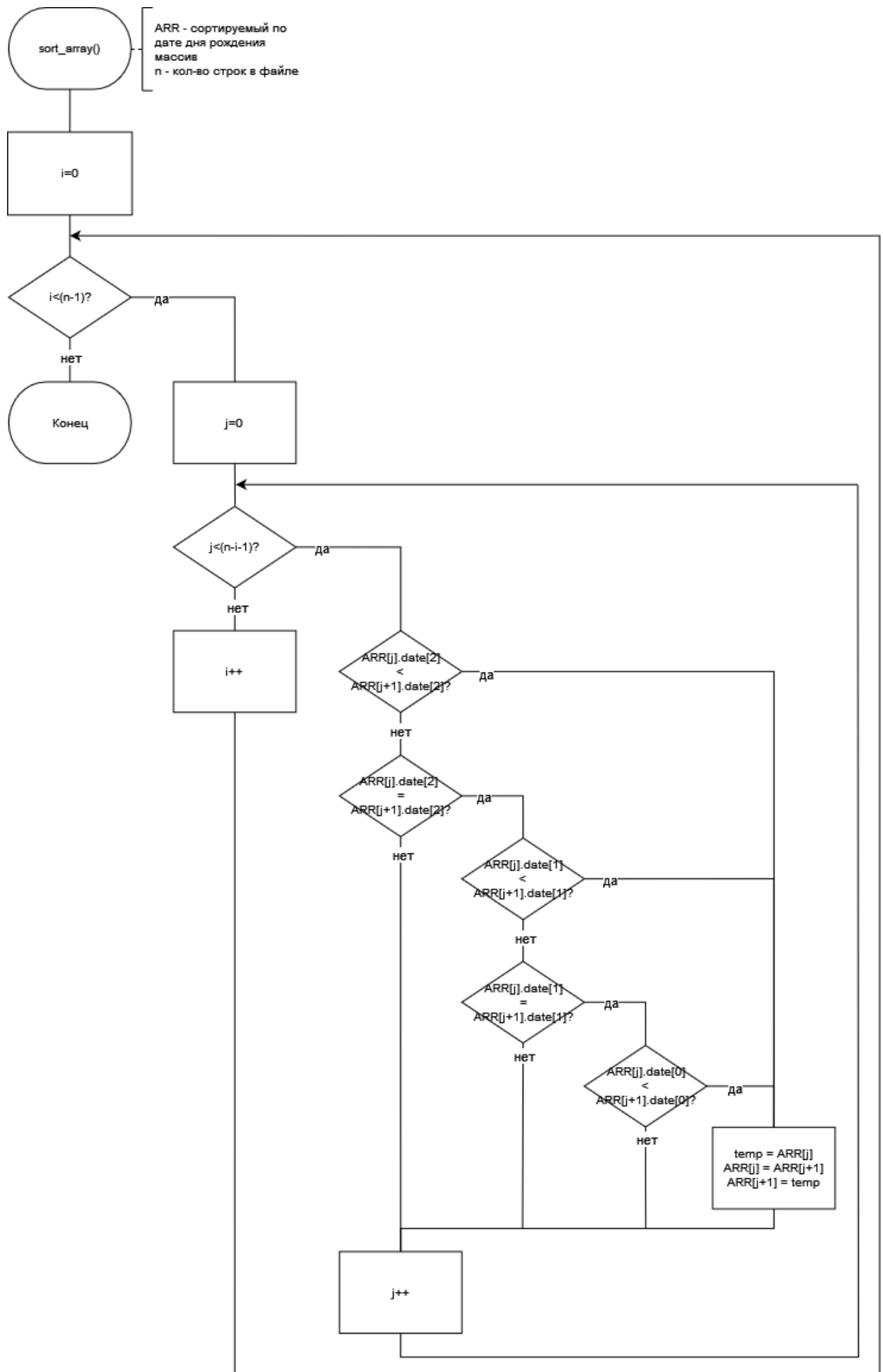


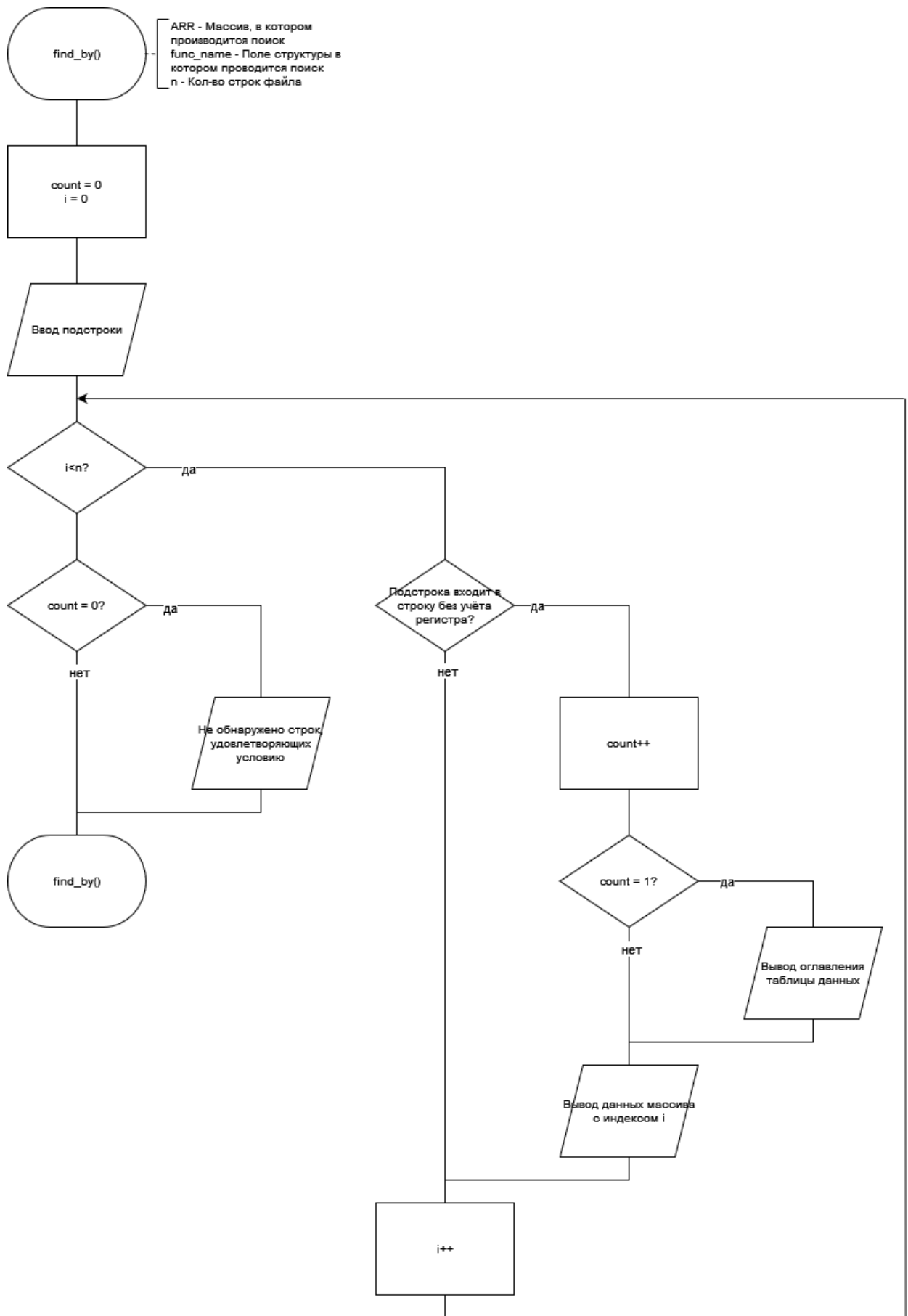












Контрольные примеры:

Пример 1:

Исходные данные:

```
Enter the separator of structure data: ;
Choose the option
0 - for EXIT program
1 - for SHOW THE DATA
2 - for ADD NEW DATA
3 - for FIND IN NAME
4 - for FIND IN DIRECTOR
Enter the option: 3
```

Your selection is FIND IN NAME

```
Enter the substring: Th
```

Результат:

Name	Director	Year	Dur	KPR	PLR	Watchdate
The Boy and Heron	Hayao Miyazaki	2023	124	7.8	8.0	17.12.2024
The Covenant	Guy Ritchie	2023	123	8.2	4.0	03.01.2024
The Banshees of Inisherin	Martin McDonagh	2022	114	7.5	6.4	05.01.2023

Пример 2:

Исходные данные:

```
Choose the option
0 - for EXIT program
1 - for SHOW THE DATA
2 - for ADD NEW DATA
3 - for FIND IN NAME
4 - for FIND IN DIRECTOR
Enter the option: 4
```

Your selection is FIND IN DIRECTOR

```
Enter the substring: mart
```

Результат:

Name	Director	Year	Dur	KPR	PLR	Watchdate
Shutter Island	Martin Scorsesse	2010	138	8.3	8.0	09.01.2024
The Banshees of Inisherin	Martin McDonagh	2022	114	7.5	6.4	05.01.2023

Пример 3:

Исходные данные:

```
Enter the separator of structure data: ;
Choose the option
0 - for EXIT program
```



```
1 - for SHOW THE DATA
2 - for ADD NEW DATA
3 - for FIND IN NAME
4 - for FIND IN DIRECTOR
Enter the option: 2
```

Your selection is ADD NEW DATA

Результат:

```
Enter the name of movie: Barbie
Enter the director of movie: Greta Gerwig
Enter the year of movie release: 2023
Enter the duration of movie: 114
Enter the KinoPoisk Rating: 6.6
Enter your Rating: 7.2
Enter the date of watch: 04 11 2023
```

Текст программы

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

enum{maxlen=128};

/* Structure */

struct movie; /* Define Structure */

typedef struct movie MOV; /* Define Structured Type */

/* Interface */

void menu(); /* Output main menu */

void print_head(); /* Output title of sheet */

void output_array(MOV *ARR[], int n); /* Output array with MOV structure */

void clear_screen(); /* Clear the console */

/* Sort, Form and other stuff */

int count_of_lines(char *filename); /* Count of lines in file */

void split_string(char *inputString, char **words, int *wordCount, char delimiter); /* Split string by separator
*/

void add_to_array(MOV *ARR[], char *filename,int n, char sep); /* Adding the data of file to array */
```

```

void new_gets(char s[], int lim); /* scan string */

void add_new_data(char *filename); /* Enter new movie */

void sort_array(MOV *ARR[], int n); /* Sort array by date of watch */

char* find_name(MOV *ARR[], int i); /* return name of movie under number [i] */

char* find_director(MOV *ARR[], int i); /* return director of movie under number [i] */

void find_by(MOV *ARR[], char *(*func_name)(MOV**, int), int n); /* find by name or director */

void clear_stream(FILE *input_stream); /* Clear the Buffer */

/* Main program */

int main(){
    int option,i;
    char sep;
    MOV **ARR=NULL;
    printf("Enter the separator of structure data: ");
    scanf("%c", &sep);

    do{
        i=count_of_lines("struct-data-win.txt");
        ARR=(MOV**)malloc(i*sizeof(MOV*));
        add_to_array(ARR,"struct-data-win.txt",i,sep);
        menu();
        scanf("%i", &option);

        switch(option){
            case 0:{
                puts("\nYour selection is EXIT");
                getchar();
                break;
            }

            case 1:{
                puts("\nYour selection is SHOW THE DATA\n");
                getchar();
                output_array(ARR,i);
                break;
            }

            case 2:{
                puts("\nYour selection is ADD NEW DATA\n");
                add_new_data("struct-data-win.txt");
                getchar();
                break;
            }

            case 3:{
                puts("\nYour selection is FIND IN NAME\n");

```

```

        getchar();
        find_by(ARR, find_name, i);
        break;
    }

    case 4:{
        puts("\nYour selection is FIND IN DIRECTOR\n");
        getchar();
        find_by(ARR, find_director, i);
        break;
    }

    default:{
        puts("\nIncorrect key");
        getchar();
    }
}

puts("\nPress ENTER to continue");
clear_stream(stdin);
clear_screen();
} while(option!=0);
return 0;
}

/* Functions */

struct movie{
    char *name; /* Name of movie */
    char *director; /* Director of movie */
    int year; /* Year of movie release */
    int duration; /* duration of movie in minutes*/
    float kpr; /* Movie rating on KinoPoisk */
    float plr; /* Movie rating on my opinion */
    int date[3]; /* Day/Month/Year of watch the movie */
};

void menu(){
    puts("Choose the option");
    puts("0 - for EXIT program");
    puts("1 - for SHOW THE DATA");
    puts("2 - for ADD NEW DATA");
    puts("3 - for FIND IN NAME");
    puts("4 - for FIND IN DIRECTOR");
    printf("Enter the option: ");
}

void print_head(){
    printf("|      %25s      |      %25s      |      %4s      |      %3s      |      %5s      |      %5s      |      %10s\n", "Name", "Director", "Year", "Dur", "KPR", "PLR", "Watchdate");
    printf("+-----+-----+-----+-----+-----+-----+-----+");
    printf("-+\n");
}

```

```

void output_array(MOV *ARR[], int n){
    int i;
    print_head();
    for(i=0;i<n;i++){
        printf("| %25s | %25s | %d | %3d | %5.1f | %5.1f | %.2d.%.2d.%.2d |\n", ARR[i]->name, ARR[i]->director,
            ARR[i]->year, ARR[i]->duration, ARR[i]->kpr, ARR[i]->plr, ARR[i]->date[0], ARR[i]->date[1],
            ARR[i]->date[2]);
    }
}

void clear_screen(){
#ifdef _WIN64
system("cls");
#else
system("clear");
#endif
}

int count_of_lines(char *filename) {
    char line[maxlen];
    int n=0;

    FILE *file = fopen(filename, "r");

    if (file!=NULL) {
        while((fgets(line,maxlen,file))!=NULL) n++;
        rewind(file);
    } else {
        printf("Error opening the file.\n");
    }
    return n;
}

void split_string(char *inputString, char **words, int *wordCount, char delimiter) {
    int wordIndex = 0, wordStart=0, wordLength=0, inWord=0, i;

    for (i = 0; i <= strlen(inputString); i++) {
        if ((inputString[i] == delimiter || inputString[i] == '\0') && (inWord==1)) {
            words[wordIndex] = (char *)malloc(wordLength + 1);
            strncpy(words[wordIndex], inputString + wordStart, wordLength);
            words[wordIndex][wordLength] = '\0';
            wordIndex++;
            inWord = 0;
        } else {
            if (inWord==0) {
                wordStart = i;
                wordLength = 1;
                inWord = 1;
            } else {
                wordLength++;
            }
        }
    }
}

```

```

        *wordCount = wordIndex;
    }

void add_to_array(MOV *ARR[], char *filename, int n, char sep){
    char line[maxlen];
    char **TMP = NULL;
    int wordCount = 0, i;

    FILE *file = fopen(filename, "r");

    if (ARR != NULL) {
        TMP = (char **)malloc(maxlen * sizeof(char *));
        for (i = 0; i < n; i++) {
            fgets(line, maxlen, file);
            split_string(line, TMP, &wordCount, sep);
            if (TMP != NULL) {
                ARR[i] = (MOV*)malloc(sizeof(MOV));
                ARR[i]->name = TMP[0];
                ARR[i]->director = TMP[1];
                ARR[i]->year = atoi(TMP[2]);
                ARR[i]->duration = atoi(TMP[3]);
                ARR[i]->kpr = atof(TMP[4]);
                ARR[i]->plr = atof(TMP[5]);
                ARR[i]->date[0] = atoi(TMP[6]);
                ARR[i]->date[1] = atoi(TMP[7]);
                ARR[i]->date[2] = atoi(TMP[8]);
            }
        }
    } else printf("Error opening the file.\n");
    fclose(file);
    for (i = 0; i < wordCount; i++) free(TMP[i]);
    free(TMP);
    sort_array(ARR,n);
}

void new_gets(char s[], int lim){
    char c;
    int i=0;
    while(((c=getchar())!='\n')&&(i<lim-1))
    {
        s[i]=c;
        i++;
    }
    s[i]='\0';
}

void add_new_data(char *filename){
    char name[maxlen],director[maxlen];
    int year,duration, d,m,y;
    float kpr, plr;
    FILE *file = fopen(filename, "a");

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

```

```

    getchar();
    new_gets(name,maxlen);
    printf("\nEnter the director of movie: ");
    new_gets(director,maxlen);
    printf("\nEnter the year of movie release: ");
    scanf("%d", &year);
    printf("\nEnter the duration of movie: ");
    scanf("%d", &duration);
    printf("\nEnter the KinoPoisk Rating: ");
    scanf("%f", &kpr);
    printf("\nEnter your Rating: ");
    scanf("%f", &plr);
    printf("\nEnter the date of watch: ");
    scanf("%d %d %d",&d,&m,&y);

    fprintf(file, "%s;%s;%d;%d;%.1f;%.1f;%2d;%2d;%d\n", name, director, year, duration, kpr, plr, d, m, y);

    fclose(file);
}

void sort_array(MOV *ARR[], int n) {
    MOV *temp;
    int i, j;
    for (i=0; i<n-1; i++) {
        for (j=0; j<n-i-1; j++) {
            if (ARR[j]->date[2] < ARR[j + 1]->date[2]) {
                temp = ARR[j];
                ARR[j] = ARR[j + 1];
                ARR[j + 1] = temp;
            } else if (ARR[j]->date[2] == ARR[j + 1]->date[2]) {
                if (ARR[j]->date[1] < ARR[j + 1]->date[1]) {
                    temp = ARR[j];
                    ARR[j] = ARR[j + 1];
                    ARR[j + 1] = temp;
                } else if (ARR[j]->date[1] == ARR[j + 1]->date[1]) {
                    if (ARR[j]->date[0] < ARR[j + 1]->date[0]) {
                        temp = ARR[j];
                        ARR[j] = ARR[j + 1];
                        ARR[j + 1] = temp;
                    }
                }
            }
        }
    }
}

char* find_name(MOV *ARR[], int i) {
    return ARR[i]->name;
}

char* find_director(MOV *ARR[], int i) {
    return ARR[i]->director;
}

```

```
void find_by(MOV *ARR[], char *(*func_name)(MOV**, int), int n){
    int i, result, count=0;
    char substring[maxlen];
    printf("Enter the substring: ");
    new_gets(substring,maxlen);
    for(i=0;i<n;i++){
        result=strncasecmp(substring,func_name(ARR,i),strlen(substring));
        if(result==0){
            count++;
            if(count==1) print_head();
            printf("| %25s | %25s | %d | %d | %5.1f | %5.1f | %.2d.%.2d.%.2d |\n", ARR[i]->name,
                ARR[i]->director, ARR[i]->year, ARR[i]->duration, ARR[i]->kpr, ARR[i]->plr, ARR[i]->date[0],
                ARR[i]->date[1], ARR[i]->date[2]);
        }
    }
    if(count==0) printf("\nThere is no name that begins with the entered substring\n");
}

void clear_stream(FILE *input_stream){
    if (!feof(input_stream)) {
        while (!feof(input_stream) && fgetc(input_stream) != '\n');
    }
}
```

Примеры выполнения программы

Исходный текстовый файл и данные:

Pulp Fiction;Quentin Tarantino;1994;154;8.5;8.9;15;01;2021
Shutter Island;Martin Scorsesse;2010;138;8.3;8.0;09;01;2024
The Boy and Heron;Hayao Miyazaki;2023;124;7.8;8.0;17;12;2024
Princess Mononoke;Hayao Miyazaki;1997;134;8.3;8.5;11;2;2024
Schindler's List;Steven Spielberg;1993;195;8.8;10.0;31;8;2020
Whiplash;Damien Chazelle;2014;107;8.4;10.0;18;1;2020
Oppenheimer;Cristopher Nolan;2023;181;8.2;8.0;18;9;2023
The Covenant;Guy Ritchie;2023;123;8.2;4.0;3;1;2024
Arrhythmia;Boris Khlebnikov;2017;112;7.2;8.2;1;12;2023
The Banshees of Inisherin;Martin McDonagh;2022;114;7.5;6.4;05;01;2023

Name	Director	Year	Dur	KPR	PLR	Watchdate
The Boy and Heron	Hayao Miyazaki	2023	124	7.8	8.0	17.12.2024
Princess Mononoke	Hayao Miyazaki	1997	134	8.3	8.5	11.02.2024
Shutter Island	Martin Scorsesse	2010	138	8.3	8.0	09.01.2024
The Covenant	Guy Ritchie	2023	123	8.2	4.0	03.01.2024
Arrhythmia	Boris Khlebnikov	2017	112	7.2	8.2	01.12.2023
Oppenheimer	Cristopher Nolan	2023	181	8.2	8.0	18.09.2023
The Banshees of Inisherin	Martin McDonagh	2022	114	7.5	6.4	05.01.2023
Pulp Fiction	Quentin Tarantino	1994	154	8.5	8.9	15.01.2021
Schindler's List	Steven Spielberg	1993	195	8.8	10.0	31.08.2020
Whiplash	Damien Chazelle	2014	107	8.4	10.0	18.01.2020

Пример 1:

```

Choose the option
0 - for EXIT program
1 - for SHOW THE DATA
2 - for ADD NEW DATA
3 - for FIND IN NAME
4 - for FIND IN DIRECTOR
Enter the option: 3

Your selection is FIND IN NAME

Enter the substring: Th
|          Name |          Director | Year | Dur | KPR | PLR | Watchdate |
+-----+-----+-----+-----+-----+-----+-----+
| The Boy and Heron | Hayao Miyazaki | 2023 | 124 | 7.8 | 8.0 | 17.12.2024 |
| The Covenant | Guy Ritchie | 2023 | 123 | 8.2 | 4.0 | 03.01.2024 |
| The Banshees of Inisherin | Martin McDonagh | 2022 | 114 | 7.5 | 6.4 | 05.01.2023 |

Press ENTER to continue

```

Пример 2:

```

Choose the option
0 - for EXIT program
1 - for SHOW THE DATA
2 - for ADD NEW DATA
3 - for FIND IN NAME
4 - for FIND IN DIRECTOR
Enter the option: 4

Your selection is FIND IN DIRECTOR

Enter the substring: mart
|          Name |          Director | Year | Dur | KPR | PLR | Watchdate |
+-----+-----+-----+-----+-----+-----+-----+
| Shutter Island | Martin Scorsesse | 2010 | 138 | 8.3 | 8.0 | 09.01.2024 |
| The Banshees of Inisherin | Martin McDonagh | 2022 | 114 | 7.5 | 6.4 | 05.01.2023 |

Press ENTER to continue

```

Пример 3:

Данные до добавления нового элемента:

Name	Director	Year	Dur	KPR	PLR	Watchdate
The Boy and Heron	Hayao Miyazaki	2023	124	7.8	8.0	17.12.2024
Princess Mononoke	Hayao Miyazaki	1997	134	8.3	8.5	11.02.2024
Shutter Island	Martin Scorsesse	2010	138	8.3	8.0	09.01.2024
The Covenant	Guy Ritchie	2023	123	8.2	4.0	03.01.2024
Arrhythmia	Boris Khlebnikov	2017	112	7.2	8.2	01.12.2023
Oppenheimer	Cristopher Nolan	2023	181	8.2	8.0	18.09.2023
The Banshees of Inisherin	Martin McDonagh	2022	114	7.5	6.4	05.01.2023
Pulp Fiction	Quentin Tarantino	1994	154	8.5	8.9	15.01.2021
Schindler's List	Steven Spielberg	1993	195	8.8	10.0	31.08.2020
Whiplash	Damien Chazelle	2014	107	8.4	10.0	18.01.2020

Добавление нового элемента:


```

Enter the separator of structure data: ;
Choose the option
0 - for EXIT program
1 - for SHOW THE DATA
2 - for ADD NEW DATA
3 - for FIND IN NAME
4 - for FIND IN DIRECTOR
Enter the option: 2

Your selection is ADD NEW DATA

Enter the name of movie: Barbie

Enter the director of movie: Greta Gerwig

Enter the year of movie release: 2023

Enter the duration of movie: 114

Enter the KinoPoisk Rating: 6.6

Enter your Rating: 7.2

Enter the date of watch: 04 11 2023

Press ENTER to continue

```

Данные после добавления нового элемента:

Name	Director	Year	Dur	KPR	PLR	Watchdate
The Boy and Heron	Hayao Miyazaki	2023	124	7.8	8.0	17.12.2024
Princess Mononoke	Hayao Miyazaki	1997	134	8.3	8.5	11.02.2024
Shutter Island	Martin Scorsesse	2010	138	8.3	8.0	09.01.2024
The Covenant	Guy Ritchie	2023	123	8.2	4.0	03.01.2024
Arrhythmia	Boris Khlebnikov	2017	112	7.2	8.2	01.12.2023
Barbie	Greta Gerwig	2023	114	6.6	7.2	04.11.2023
Oppenheimer	Cristopher Nolan	2023	181	8.2	8.0	18.09.2023
The Banshees of Inisherin	Martin McDonagh	2022	114	7.5	6.4	05.01.2023
Pulp Fiction	Quentin Tarantino	1994	154	8.5	8.9	15.01.2021
Schindler's List	Steven Spielberg	1993	195	8.8	10.0	31.08.2020
Whiplash	Damien Chazelle	2014	107	8.4	10.0	18.01.2020

Выводы.

В результате выполнения работы изучена и освоена работа с указателями на структуры и функции.