

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

ОТЧЕТ
по лабораторной работе № 8
по дисциплине «Программирование»
ТЕМА: Линейные односвязные списки

Студент гр. 3311

Баймухамедов Р.Р

Преподаватель

Хахаев И.А.

Санкт-Петербург

2024

Цель работы

Целью работы является изучение и освоение работы с линейными односвязными списками.

Задание (вариант 9)

Для выбранной предметной области создать односвязный линейный список 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 односвязного линейного списка структур при чтении из файла (текст с разделителями — CSV), и разработать подалгоритм удаления элементов с заданным содержимым указанного информационного поля из односвязного списка. При отсутствии таких элементов в списке вывести соответствующее сообщение.

Постановка задачи и описание решения

Для выполнения данной лабораторной работы необходимо разработать программу, которая будет читать данные из файла и записывать их в линейный односвязный список. Предметная область будет фильмы: название фильма, режиссёр, год выхода, длительность, рейтинг КиноПоиска, личный рейтинг и дата просмотра. В качестве информационных полей, в которых будет производиться поиск для удаления элементов были выбраны поля `year` (Год выхода фильма, тип `int`) и `plr` (Личный рейтинг фильма, тип `float`) Для достижения этой цели необходимо выполнить следующие шаги:

1. Запросить у пользователя символ-разделитель.

2. Прочитывать данные строки файла, разделенные ранее введенным символом-разделителем, и создавать из прочтенных данных линейный односвязный список, до тех пор, пока данные всех строк не будут занесены в список.
3. Запросить у пользователя по какому полю структуры будет проводиться поиск для удаления элементов.
4. Запросить у пользователя значение, по которому будет проводиться поиск элементов для удаления в ранее выбранном поле.
5. Удалить элементы, удовлетворяющие условию, из линейного односвязного списка.

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	make_head	HD	Инициализация головы списка
3	create_movie	MOV	Создание элемента списка
4	add_node	void	Привязка нового элемента к списку
5	delete_node	void	Удаление элемента из списка
6	reset_to_original_data	void	Сброс списка до исходного значения
7	menu	void	Выводит варианты выбора пользователя
8	print_head	void	Вывод оглавления таблицы данных
9	output_list	void	Вывод списка
10	split_string	void	Разделяет строку по символу-разделителю и записывает элементы строки в массив
11	add_to_list	void	Чтение из файла и запись элементов строк в линейный односвязный список
12	find_year	float	Возвращает год выхода фильма в списке под переданным указателем элемента
13	find_plr	float	Возвращает личный рейтинг фильма в списке под переданным указателем элемента
14	find_by	void	Ввод значения и проверка значений выбранного поля структуры на их совпадение, удаление элементов списка при удовлетворении условия
15	clear_stream	void	Очистка входного буфера

Описание переменных

int main()

№	Переменная	Тип	Назначение
1	*ph	HD (struct head)	Голова списка, которая хранит в себе указатель на первый и последний элемент списка, а также счётчик элементов
3	option	int	Опция, выбираемая пользователем
4	sep	char	Символ-разделитель

struct movie

№	Название	Тип	Назначение
2	*name	char	Название фильма
3	*director	char	Имя и фамилия режиссёра фильма
4	year	int	Год выхода фильма
5	duration	int	Длительность фильма
6	kpr	float	Оценка фильма на КиноПоиске
7	plr	float	Личная оценка фильма
8	date[3]	int	Дата просмотра фильма
9	*next	MOV (struct movie)	Указатель на следующий элемент списка

struct head

№	Название	Тип	Назначение
1	cnt	int	Счётчик элементов
2	*first	MOV (struct movie)	Указатель на первый элемент списка
3	*last	MOV (struct movie)	Указатель на последний элемент списка

HD *make_head()

№	Название	Тип	Назначение
---	----------	-----	------------

1	*ph	HD	Голова линейного односвязного списка
---	-----	----	--------------------------------------

```
MOV *create_movie(char *movie_name, char *movie_director, int
movie_year, int movie_duration, float movie_kpr, float movie_plr, int
watch_date[3])
```

№	Название	Тип	Назначение
1	*new_movie	MOV	Новый элемент списка
2	*name	char	указатель на строку для хранения названия фильма
3	*director	char	указатель на строку для хранения режиссёра фильма

```
void add_node(HD *ph, MOV *new_node, MOV *current_node)
```

№	Название	Тип	Назначение
1	n	int	Счётчик элементов в голове

```
void delete_node(HD *ph, MOV *current_node)
```

№	Название	Тип	Назначение
1	*q	MOV	Указатель на текущий элемент
2	*ql	MOV	Указатель на последний элемент

```
void reset_to_original_data(char *filename, HD *ph, char sep)
```

№	Название	Тип	Назначение
1	*current	MOV	Указатель на текущий элемент
2	*temp	MOV	Указатель, временно хранящий элемент списка

```
void output_list(HD *ph)
```

№	Название	Тип	Назначение
1	*current	MOV	Указатель на текущий элемент

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_list(char *filename, HD *ph, char sep)

№	Переменная	Тип	Назначение
1	*words[9]	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, на которые была разделена строка
5	date[3]	int	Массив для хранения даты просмотра фильма
6	*new_movie	MOV	Указатель на структуру MOV,

			представляющую новый фильм
--	--	--	-------------------------------

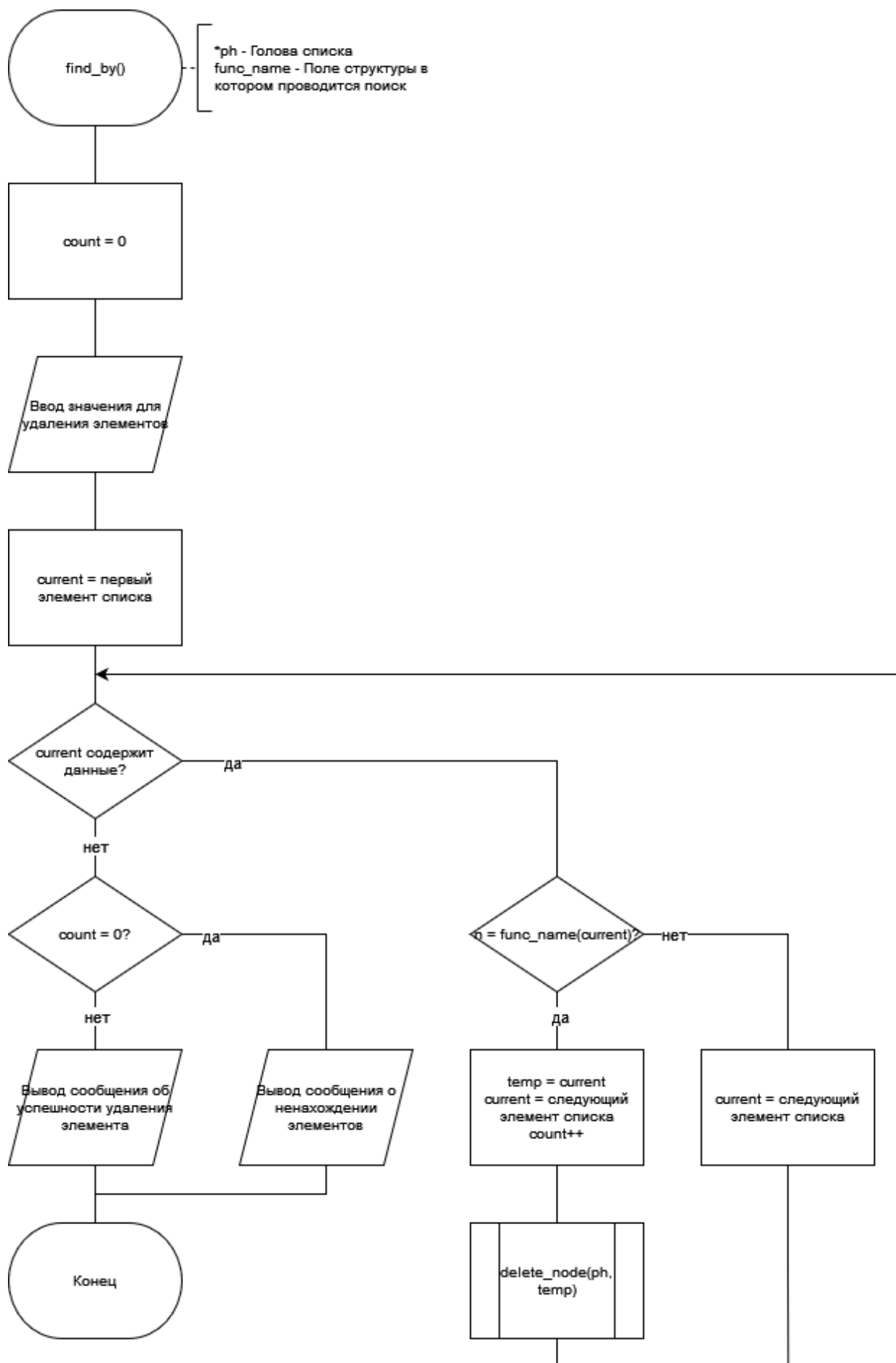
void find_by(HD *ph, float (*func_name)(MOV*))

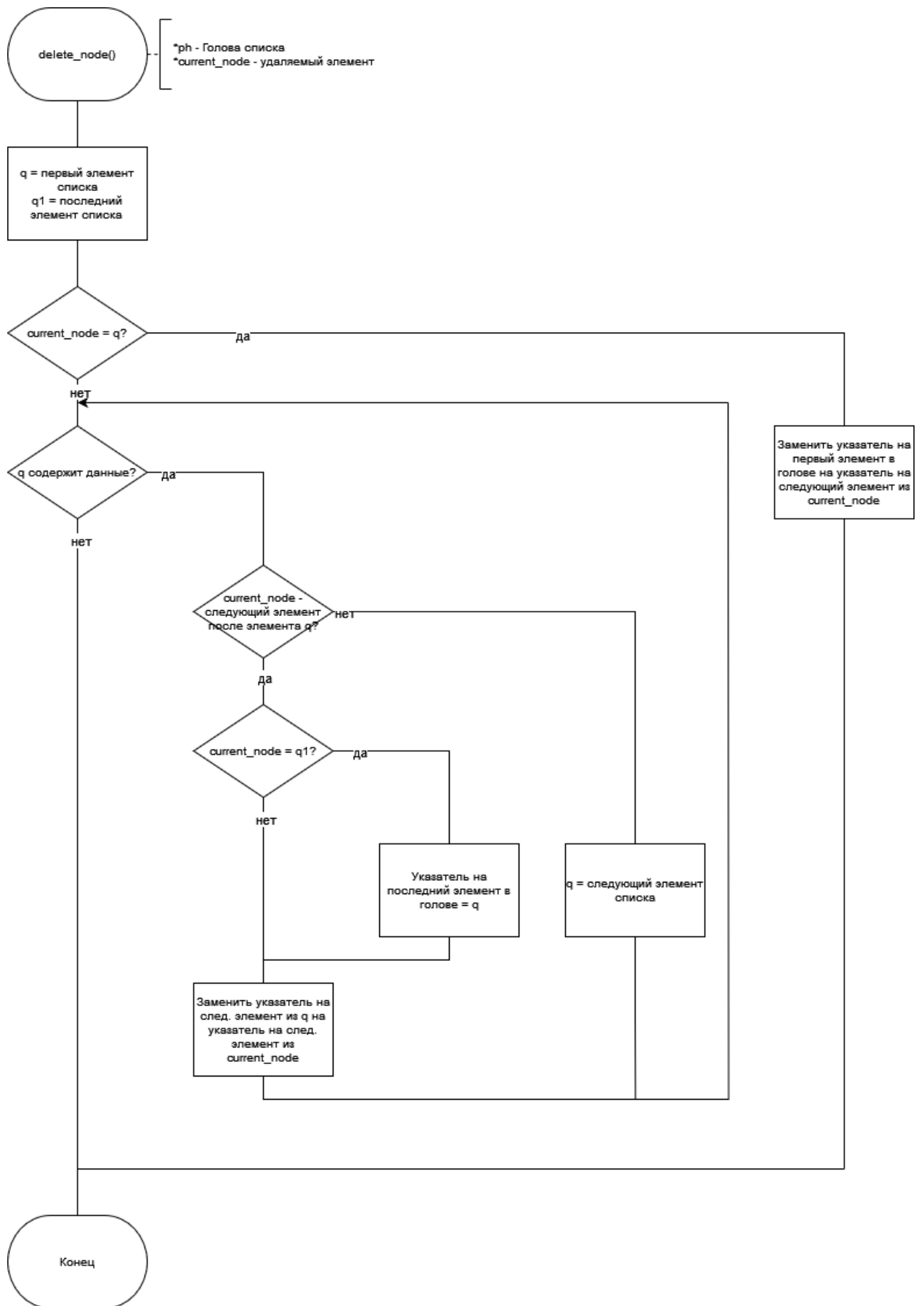
№	Переменная	Тип	Назначение
1	*current	MOV	Указатель на текущий элемент списка
2	*temp	MOV	Указатель, временно хранящий элемент списка
3	count	int	Количество элементов, удовлетворяющих условию
4	n	float	Вводимое пользователем значение, по которому происходит поиск элементов списка для удаления

void clear_stream(FILE *input_stream)

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

Схемы алгоритмов





Пример 1:

Исходные данные:

Enter the separator of structure data: ;
Choose the option
0 - for EXIT program
1 - for SHOW THE DATA
2 - for DELETE THE DATA BY YEAR
3 - for DELETE THE DATA BY PLR
4 - for RESET TO ORIGINAL DATA
Enter the option: 2

Your selection is DELETE THE DATA BY YEAR

ID	Name	Director	Year	Dur	KPR	PLR	Watchdate
1	Pulp Fiction	Quentin Tarantino	1994	154	8.5	8.9	15.01.2021
2	Shutter Island	Martin Scorsesse	2010	138	8.3	8.0	09.01.2024
3	The Boy and Heron	Hayao Miyazaki	2023	124	7.8	8.0	17.12.2024
4	Princess Mononoke	Hayao Miyazaki	1997	134	8.3	8.5	11.02.2024
5	Schindler's List	Steven Spielberg	1993	195	8.8	10.0	31.08.2020
6	Whiplash	Damien Chazelle	2014	107	8.4	10.0	18.01.2020
7	Oppenheimer	Cristopher Nolan	2023	181	8.2	8.0	18.09.2023
8	The Covenant	Guy Ritchie	2023	123	8.2	4.0	03.01.2024
9	Arrhythmia	Boris Khlebnikov	2017	112	7.2	8.2	01.12.2023
10	The Banshees of Inisherin	Martin McDonagh	2022	114	7.5	6.4	05.01.2023
11	Barbie	Greta Gerwig	2023	114	6.6	7.2	04.11.2023
12	Yellow Submarine	George Dunning	1968	89	8.0	8.1	03.11.2023
13	The Lighthouse	Robert Eggers	2019	109	7.1	9.5	05.02.2024

Enter a field value to remove elements of list with the same field value: 2023

All elements, which include entered value in chosen field were successfully removed from list

Результат:

ID	Name	Director	Year	Dur	KPR	PLR	Watchdate
1	Pulp Fiction	Quentin Tarantino	1994	154	8.5	8.9	15.01.2021
2	Shutter Island	Martin Scorsesse	2010	138	8.3	8.0	09.01.2024
4	Princess Mononoke	Hayao Miyazaki	1997	134	8.3	8.5	11.02.2024
5	Schindler's List	Steven Spielberg	1993	195	8.8	10.0	31.08.2020
6	Whiplash	Damien Chazelle	2014	107	8.4	10.0	18.01.2020
9	Arrhythmia	Boris Khlebnikov	2017	112	7.2	8.2	01.12.2023
10	The Banshees of Inisherin	Martin McDonagh	2022	114	7.5	6.4	05.01.2023
12	Yellow Submarine	George Dunning	1968	89	8.0	8.1	03.11.2023
13	The Lighthouse	Robert Eggers	2019	109	7.1	9.5	05.02.2024

Пример 2:

Исходные данные:

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

1 - for SHOW THE DATA
2 - for DELETE THE DATA BY YEAR
3 - for DELETE THE DATA BY PLR
4 - for RESET TO ORIGINAL DATA
Enter the option: 3

Your selection is DELETE THE DATA BY PLR

ID	Name	Director	Year	Dur	KPR	PLR	Watchdate
1	Pulp Fiction	Quentin Tarantino	1994	154	8.5	8.9	15.01.2021
2	Shutter Island	Martin Scorsesse	2010	138	8.3	8.0	09.01.2024
3	The Boy and Heron	Hayao Miyazaki	2023	124	7.8	8.0	17.12.2024
4	Princess Mononoke	Hayao Miyazaki	1997	134	8.3	8.5	11.02.2024
5	Schindler's List	Steven Spielberg	1993	195	8.8	10.0	31.08.2020
6	Whiplash	Damien Chazelle	2014	107	8.4	10.0	18.01.2020
7	Oppenheimer	Cristopher Nolan	2023	181	8.2	8.0	18.09.2023
8	The Covenant	Guy Ritchie	2023	123	8.2	4.0	03.01.2024
9	Arrhythmia	Boris Khlebnikov	2017	112	7.2	8.2	01.12.2023
10	The Banshees of Inisherin	Martin McDonagh	2022	114	7.5	6.4	05.01.2023
11	Barbie	Greta Gerwig	2023	114	6.6	7.2	04.11.2023
12	Yellow Submarine	George Dunning	1968	89	8.0	8.1	03.11.2023
13	The Lighthouse	Robert Eggers	2019	109	7.1	9.5	05.02.2024

Enter a field value to remove elements of list with the same field value: 8.0

All elements, which include entered value in chosen field were successfully removed from list

Результат:

ID	Name	Director	Year	Dur	KPR	PLR	Watchdate
1	Pulp Fiction	Quentin Tarantino	1994	154	8.5	8.9	15.01.2021
4	Princess Mononoke	Hayao Miyazaki	1997	134	8.3	8.5	11.02.2024
5	Schindler's List	Steven Spielberg	1993	195	8.8	10.0	31.08.2020
6	Whiplash	Damien Chazelle	2014	107	8.4	10.0	18.01.2020
8	The Covenant	Guy Ritchie	2023	123	8.2	4.0	03.01.2024
9	Arrhythmia	Boris Khlebnikov	2017	112	7.2	8.2	01.12.2023
10	The Banshees of Inisherin	Martin McDonagh	2022	114	7.5	6.4	05.01.2023
11	Barbie	Greta Gerwig	2023	114	6.6	7.2	04.11.2023
12	Yellow Submarine	George Dunning	1968	89	8.0	8.1	03.11.2023
13	The Lighthouse	Robert Eggers	2019	109	7.1	9.5	05.02.2024

Пример 3: (В дополнении к 2 примеру)

Исходные данные:

Choose the option
0 - for EXIT program
1 - for SHOW THE DATA
2 - for DELETE THE DATA BY YEAR
3 - for DELETE THE DATA BY PLR
4 - for RESET TO ORIGINAL DATA
Enter the option: 2

Your selection is DELETE THE DATA BY YEAR

ID	Name	Director	Year	Dur	KPR	PLR	Watchdate
+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+
1	Pulp Fiction	Quentin Tarantino	1994	154	8.5	8.9	15.01.2021
4	Princess Mononoke	Hayao Miyazaki	1997	134	8.3	8.5	11.02.2024
5	Schindler's List	Steven Spielberg	1993	195	8.8	10.0	31.08.2020
6	Whiplash	Damien Chazelle	2014	107	8.4	10.0	18.01.2020
8	The Covenant	Guy Ritchie	2023	123	8.2	4.0	03.01.2024
9	Arrhythmia	Boris Khlebnikov	2017	112	7.2	8.2	01.12.2023
10	The Banshees of Inisherin	Martin McDonagh	2022	114	7.5	6.4	05.01.2023
11	Barbie	Greta Gerwig	2023	114	6.6	7.2	04.11.2023
12	Yellow Submarine	George Dunning	1968	89	8.0	8.1	03.11.2023
13	The Lighthouse	Robert Eggers	2019	109	7.1	9.5	05.02.2024

Enter a field value to remove elements of list with the same field value: 2023

All elements, which include entered value in chosen field were successfully removed from list

Результат:

ID	Name	Director	Year	Dur	KPR	PLR	Watchdate
+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+
1	Pulp Fiction	Quentin Tarantino	1994	154	8.5	8.9	15.01.2021
4	Princess Mononoke	Hayao Miyazaki	1997	134	8.3	8.5	11.02.2024
5	Schindler's List	Steven Spielberg	1993	195	8.8	10.0	31.08.2020
6	Whiplash	Damien Chazelle	2014	107	8.4	10.0	18.01.2020
9	Arrhythmia	Boris Khlebnikov	2017	112	7.2	8.2	01.12.2023
10	The Banshees of Inisherin	Martin McDonagh	2022	114	7.5	6.4	05.01.2023
12	Yellow Submarine	George Dunning	1968	89	8.0	8.1	03.11.2023
13	The Lighthouse	Robert Eggers	2019	109	7.1	9.5	05.02.2024

Текст программы

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#define maxlen 128

/* Structure */

struct movie; /* Define structure of movie */

struct head; /* Define structure of List Head*/

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

typedef struct head HD; /* Define Structured Type of List Head*/

/* Node and List */

HD *make_head(); /* Head initialization */

MOV *create_movie(char *movie_name, char *movie_director, int movie_year, int movie_duration, float movie_kpr,
float movie_plr, int watch_date[3]); /* Node Initialization*/
```

```

void add_node(HD *ph, MOV *new_node, MOV *current_node); /* Add new Node to List */

void delete_node(HD *ph, MOV *current_node); /* Delete selected Node */

void reset_to_original_data(char *filename, HD *ph, char sep); /* Reset List to Original Data */

/* Add last right now in useless, but function already written and saved in test.c */

/* Interface */

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

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

void output_list(HD *head); /* Output array with MOV structure */

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

/* Sort, Form and other Stuff */

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

void add_to_list(char *filename, HD *ph, char sep); /* Adding the data of file to array */

float find_year(MOV *current_node);

float find_plr(MOV *current_node);

void find_by(HD *ph, float (*func_name)(MOV*)); /* Find by year or PLR */

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

/* Main Program */

int main() {
    char sep;
    int option;
    HD *ph;

    printf("Enter the separator of structure data: ");
    scanf("%c", &sep);
    ph=make_head();
    add_to_list("struct-data-win.txt", ph,sep);

    do{
        menu();
        scanf("%i", &option);

        switch(option){
            case 0:{
                puts("\nYour selection is EXIT");
            }
        }
    } while(option != 0);
}

```

```

        getchar();
        break;
    }

    case 1:{
        puts("\nYour selection is SHOW THE DATA\n");
        getchar();
        output_list(ph);
        break;
    }

    case 2:{
        puts("\nYour selection is DELETE THE DATA BY YEAR\n");
        output_list(ph);
        find_by(ph,find_year);
        getchar();
        break;
    }

    case 3:{
        puts("\nYour selection is DELETE THE DATA BY PLR\n");
        output_list(ph);
        find_by(ph,find_plr);
        getchar();
        break;
    }

    case 4:{
        puts("\nYour selection is RESET TO ORIGINAL DATA\n");
        reset_to_original_data("struct-data-win.txt",ph,sep);
        printf("Reset to original data is completed\n");
        getchar();
        break;
    }

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

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

return 0;
}

struct movie{
    int id; /* ID of element */
    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 */
    struct movie *next; /* Link to next node */
};

struct head{
    int cnt; /* Counter of all elements that ever exist */
    struct movie *first;
    struct movie *last;
};

HD *make_head(){
    HD *ph=NULL; /* Define and init Head */
    ph=(HD*)malloc(sizeof(HD));
    ph->cnt=0;
    ph->first=NULL;
    ph->last=NULL;
    return ph;
}

MOV *create_movie(char *movie_name, char *movie_director, int movie_year, int movie_duration, float movie_kpr,
    float movie_plr, int watch_date[3]){
    MOV *new_movie = NULL; /* Pointer to new node */
    char *name = NULL;
    char *director = NULL;

    new_movie = (MOV*)malloc(sizeof(MOV));
    name = (char*)malloc((strlen(movie_name) + 1) * sizeof(char));
    director = (char*)malloc((strlen(movie_director) + 1) * sizeof(char));

    if (new_movie && name && director){ /* Data is not empty */
        new_movie->id = 1; /* Setting node ID to 1 */
        new_movie->name = name;
        new_movie->director = director;
        new_movie->year = movie_year;
        new_movie->duration = movie_duration;
        new_movie->kpr = movie_kpr;
        new_movie->plr = movie_plr;
        memcpy(name, movie_name, strlen(movie_name) + 1);
        memcpy(director, movie_director, strlen(movie_director) + 1);
        memcpy(new_movie->date, watch_date, sizeof(new_movie->date));
        new_movie->next = NULL;
    }

    return new_movie; /* return adres of node */
}

void add_node(HD *ph, MOV *new_node, MOV *current_node){
    int n;
    if(ph&&new_node){
        n=ph->cnt+1;

```

```

        if (current_node == NULL) { /* Add first node of list */
            ph->first = new_node;
            ph->last = new_node;
        } else{
            current_node->next=new_node;
            ph->last=new_node;
        }

        new_node->id=n;
        ph->cnt=n;
    }
}

```

```

void delete_node(HD *ph, MOV *current_node){
    MOV *q, *q1;

    q=ph->first;
    q1=ph->last;
    if(current_node==q){
        ph->first=current_node->next;
        current_node->next=NULL;
        free(current_node);
    }
    else{
        while (q!=NULL) {
            if (q->next==current_node) {
                if (current_node==q1) ph->last=q;
                q->next=current_node->next;
                current_node->next=NULL;
                free(current_node);
            } else q=q->next;
        }
    }
}

```

```

void reset_to_original_data(char *filename, HD *ph, char sep) {
    MOV *current = ph->first;
    MOV *temp;

    ph->cnt=0;

    while (current != NULL) {
        temp = current;
        current = current->next;
        free(temp);
    }

    ph->first = NULL;
    ph->last = NULL;

    add_to_list(filename, ph, sep);
}

```



```
void menu(){
    puts("Choose the option");
    puts("0 - for EXIT program");
    puts("1 - for SHOW THE DATA");
    puts("2 - for DELETE THE DATA BY YEAR");
    puts("3 - for DELETE THE DATA BY PLR");
    puts("4 - for RESET TO ORIGINAL DATA");
    printf("Enter the option: ");
}

void print_head(){
    printf("| %2s | %25s | %25s | %4s | %3s | %5s | %5s | %10s\n", "ID", "Name", "Director", "Year", "Dur", "KPR", "PLR", "Watchdate");
    printf("+---+-----+-----+----+--+-----+-----+-----+-----+\n");
}

void output_list(HD *ph) {
    MOV *current;

    current = ph->first;
    print_head();

    while (current!=NULL) {
        printf("| %2d | %25s | %25s | %d | %3d | %5.1f | %5.1f | %.2d%.2d.%d |\n", current->id, current->name,
            current->director, current->year, current->duration, current->kpr, current->plr, current->date[0],
            current->date[1], current->date[2]);
        current = current->next;
    }
}

void clear_screen(){
#ifdef _WIN32 || defined(_WIN64)
    system("cls");
#else
    system("clear");
#endif
}

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_list(char *filename, HD *ph, char sep){
    char line[maxlen], *words[9];
    int wordCount, date[3], i;
    MOV *new_movie;
    FILE *file = fopen(filename, "r");
    if (file == NULL) printf("Error opening file.\n");

    while (fgets(line, sizeof(line), file)){
        split_string(line, words, &wordCount, sep);

        if (wordCount == 9){
            char *name = words[0];
            char *director = words[1];
            int year = atoi(words[2]);
            int duration = atoi(words[3]);
            float kpr = atof(words[4]);
            float plr = atof(words[5]);
            date[0] = atoi(words[6]);
            date[1] = atoi(words[7]);
            date[2] = atoi(words[8]);

            new_movie = create_movie(name, director, year, duration, kpr, plr, date);
            add_node(ph, new_movie, ph->last);
        }
        else printf("Invalid number of attributes in line: %s\n", line);

        for (i = 0; i < wordCount; i++){
            free(words[i]);
        }
    }
    fclose(file);
}

float find_year(MOV *current_node){
    return current_node->year;
}

float find_plr(MOV *current_node){
    return current_node->plr;
}

void find_by(HD *ph, float (*func_name)(MOV*)) {
    MOV *current, *temp;
    int count=0;

```

```

float n;
printf("\nEnter a field value to remove elements of list with the same field value: ");
scanf("%f", &n);

current = ph->first;
while(current != NULL) {
    if(n == func_name(current)) {
        temp = current;
        current = current->next;
        delete_node(ph, temp);
        count++;
    } else {
        current = current->next;
    }
}
if (count==0){
    printf("\nThere is no field that include entered value\n");
} else printf("\nAll elements, which include entered value in chosen field were successfully removed from
list\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
Barbie;Greta Gerwig;2023;114;6.6;7.2; 4;11;2023
Yellow Submarine;George Dunning;1968;89;8.0;8.1; 3;11;2023
The Lighthouse;Robert Eggers;2019;109;7.1;9.5; 5; 2;2024

```

ID	Name	Director	Year	Dur	KPR	PLR	Watchdate
1	Pulp Fiction	Quentin Tarantino	1994	154	8.5	8.9	15.01.2021
2	Shutter Island	Martin Scorsesse	2010	138	8.3	8.0	09.01.2024
3	The Boy and Heron	Hayao Miyazaki	2023	124	7.8	8.0	17.12.2024
4	Princess Mononoke	Hayao Miyazaki	1997	134	8.3	8.5	11.02.2024
5	Schindler's List	Steven Spielberg	1993	195	8.8	10.0	31.08.2020
6	Whiplash	Damien Chazelle	2014	107	8.4	10.0	18.01.2020
7	Oppenheimer	Cristopher Nolan	2023	181	8.2	8.0	18.09.2023
8	The Covenant	Guy Ritchie	2023	123	8.2	4.0	03.01.2024
9	Arrhythmia	Boris Khlebnikov	2017	112	7.2	8.2	01.12.2023
10	The Banshees of Inisherin	Martin McDonagh	2022	114	7.5	6.4	05.01.2023
11	Barbie	Greta Gerwig	2023	114	6.6	7.2	04.11.2023
12	Yellow Submarine	George Dunning	1968	89	8.0	8.1	03.11.2023
13	The Lighthouse	Robert Eggers	2019	109	7.1	9.5	05.02.2024

Пример 1:

```

1 - for SHOW THE DATA
2 - for DELETE THE DATA BY YEAR
3 - for DELETE THE DATA BY PLR
4 - for RESET TO ORIGINAL DATA
Enter the option: 2

Your selection is DELETE THE DATA BY YEAR

ID | Name | Director | Year | Dur | KPR | PLR | Watchdate |
+---+-----+-----+-----+-----+-----+-----+
1 | Pulp Fiction | Quentin Tarantino | 1994 | 154 | 8.5 | 8.9 | 15.01.2021 |
2 | Shutter Island | Martin Scorsesse | 2010 | 138 | 8.3 | 8.0 | 09.01.2024 |
3 | The Boy and Heron | Hayao Miyazaki | 2023 | 124 | 7.8 | 8.0 | 17.12.2024 |
4 | Princess Mononoke | Hayao Miyazaki | 1997 | 134 | 8.3 | 8.5 | 11.02.2024 |
5 | Schindler's List | Steven Spielberg | 1993 | 195 | 8.8 | 10.0 | 31.08.2020 |
6 | Whiplash | Damien Chazelle | 2014 | 107 | 8.4 | 10.0 | 18.01.2020 |
7 | Oppenheimer | Cristopher Nolan | 2023 | 181 | 8.2 | 8.0 | 18.09.2023 |
8 | The Covenant | Guy Ritchie | 2023 | 123 | 8.2 | 4.0 | 03.01.2024 |
9 | Arrhythmia | Boris Khlebnikov | 2017 | 112 | 7.2 | 8.2 | 01.12.2023 |
10 | The Banshees of Inisherin | Martin McDonagh | 2022 | 114 | 7.5 | 6.4 | 05.01.2023 |
11 | Barbie | Greta Gerwig | 2023 | 114 | 6.6 | 7.2 | 04.11.2023 |
12 | Yellow Submarine | George Dunning | 1968 | 89 | 8.0 | 8.1 | 03.11.2023 |
13 | The Lighthouse | Robert Eggers | 2019 | 109 | 7.1 | 9.5 | 05.02.2024 |

Enter a field value to remove elements of list with the same field value: 2023

All elements, which include entered value in chosen field were successfully removed from list

Press ENTER to continue

```

```

Choose the option
0 - for EXIT program
1 - for SHOW THE DATA
2 - for DELETE THE DATA BY YEAR
3 - for DELETE THE DATA BY PLR
4 - for RESET TO ORIGINAL DATA
Enter the option: 1

Your selection is SHOW THE DATA

ID | Name | Director | Year | Dur | KPR | PLR | Watchdate |
+---+-----+-----+-----+-----+-----+-----+
1 | Pulp Fiction | Quentin Tarantino | 1994 | 154 | 8.5 | 8.9 | 15.01.2021 |
2 | Shutter Island | Martin Scorsesse | 2010 | 138 | 8.3 | 8.0 | 09.01.2024 |
4 | Princess Mononoke | Hayao Miyazaki | 1997 | 134 | 8.3 | 8.5 | 11.02.2024 |
5 | Schindler's List | Steven Spielberg | 1993 | 195 | 8.8 | 10.0 | 31.08.2020 |
6 | Whiplash | Damien Chazelle | 2014 | 107 | 8.4 | 10.0 | 18.01.2020 |
9 | Arrhythmia | Boris Khlebnikov | 2017 | 112 | 7.2 | 8.2 | 01.12.2023 |
10 | The Banshees of Inisherin | Martin McDonagh | 2022 | 114 | 7.5 | 6.4 | 05.01.2023 |
12 | Yellow Submarine | George Dunning | 1968 | 89 | 8.0 | 8.1 | 03.11.2023 |
13 | The Lighthouse | Robert Eggers | 2019 | 109 | 7.1 | 9.5 | 05.02.2024 |

Press ENTER to continue

```

Пример 2:

```

Choose the option
0 - for EXIT program
1 - for SHOW THE DATA
2 - for DELETE THE DATA BY YEAR
3 - for DELETE THE DATA BY PLR
4 - for RESET TO ORIGINAL DATA
Enter the option: 3

Your selection is DELETE THE DATA BY PLR

```

ID	Name	Director	Year	Dur	KPR	PLR	Watchdate
1	Pulp Fiction	Quentin Tarantino	1994	154	8.5	8.9	15.01.2021
2	Shutter Island	Martin Scorsesse	2010	138	8.3	8.0	09.01.2024
3	The Boy and Heron	Hayao Miyazaki	2023	124	7.8	8.0	17.12.2024
4	Princess Mononoke	Hayao Miyazaki	1997	134	8.3	8.5	11.02.2024
5	Schindler's List	Steven Spielberg	1993	195	8.8	10.0	31.08.2020
6	Whiplash	Damien Chazelle	2014	107	8.4	10.0	18.01.2020
7	Oppenheimer	Cristopher Nolan	2023	181	8.2	8.0	18.09.2023
8	The Covenant	Guy Ritchie	2023	123	8.2	4.0	03.01.2024
9	Arrhythmia	Boris Khlebnikov	2017	112	7.2	8.2	01.12.2023
10	The Banshees of Inisherin	Martin McDonagh	2022	114	7.5	6.4	05.01.2023
11	Barbie	Greta Gerwig	2023	114	6.6	7.2	04.11.2023
12	Yellow Submarine	George Dunning	1968	89	8.0	8.1	03.11.2023
13	The Lighthouse	Robert Eggers	2019	109	7.1	9.5	05.02.2024

```

Enter a field value to remove elements of list with the same field value: 8.0

All elements, which include entered value in chosen field were successfully removed from list

Press ENTER to continue

```

```

Choose the option
0 - for EXIT program
1 - for SHOW THE DATA
2 - for DELETE THE DATA BY YEAR
3 - for DELETE THE DATA BY PLR
4 - for RESET TO ORIGINAL DATA
Enter the option: 1

Your selection is SHOW THE DATA

```

ID	Name	Director	Year	Dur	KPR	PLR	Watchdate
1	Pulp Fiction	Quentin Tarantino	1994	154	8.5	8.9	15.01.2021
4	Princess Mononoke	Hayao Miyazaki	1997	134	8.3	8.5	11.02.2024
5	Schindler's List	Steven Spielberg	1993	195	8.8	10.0	31.08.2020
6	Whiplash	Damien Chazelle	2014	107	8.4	10.0	18.01.2020
8	The Covenant	Guy Ritchie	2023	123	8.2	4.0	03.01.2024
9	Arrhythmia	Boris Khlebnikov	2017	112	7.2	8.2	01.12.2023
10	The Banshees of Inisherin	Martin McDonagh	2022	114	7.5	6.4	05.01.2023
11	Barbie	Greta Gerwig	2023	114	6.6	7.2	04.11.2023
12	Yellow Submarine	George Dunning	1968	89	8.0	8.1	03.11.2023
13	The Lighthouse	Robert Eggers	2019	109	7.1	9.5	05.02.2024

```

Press ENTER to continue

```

Пример 3:

```

Choose the option
0 - for EXIT program
1 - for SHOW THE DATA
2 - for DELETE THE DATA BY YEAR
3 - for DELETE THE DATA BY PLR
4 - for RESET TO ORIGINAL DATA
Enter the option: 2

Your selection is DELETE THE DATA BY YEAR

| ID | Name | Director | Year | Dur | KPR | PLR | Watchdate |
|----|-----|-----|-----|-----|-----|-----|-----|
| 1 | Pulp Fiction | Quentin Tarantino | 1994 | 154 | 8.5 | 8.9 | 15.01.2021 |
| 4 | Princess Mononoke | Hayao Miyazaki | 1997 | 134 | 8.3 | 8.5 | 11.02.2024 |
| 5 | Schindler's List | Steven Spielberg | 1993 | 195 | 8.8 | 10.0 | 31.08.2020 |
| 6 | Whiplash | Damien Chazelle | 2014 | 107 | 8.4 | 10.0 | 18.01.2020 |
| 8 | The Covenant | Guy Ritchie | 2023 | 123 | 8.2 | 4.0 | 03.01.2024 |
| 9 | Arrhythmia | Boris Khlebnikov | 2017 | 112 | 7.2 | 8.2 | 01.12.2023 |
| 10 | The Banshees of Inisherin | Martin McDonagh | 2022 | 114 | 7.5 | 6.4 | 05.01.2023 |
| 11 | Barbie | Greta Gerwig | 2023 | 114 | 6.6 | 7.2 | 04.11.2023 |
| 12 | Yellow Submarine | George Dunning | 1968 | 89 | 8.0 | 8.1 | 03.11.2023 |
| 13 | The Lighthouse | Robert Eggers | 2019 | 109 | 7.1 | 9.5 | 05.02.2024 |

Enter a field value to remove elements of list with the same field value: 2023

All elements, which include entered value in chosen field were successfully removed from list

Press ENTER to continue

```

```

Choose the option
0 - for EXIT program
1 - for SHOW THE DATA
2 - for DELETE THE DATA BY YEAR
3 - for DELETE THE DATA BY PLR
4 - for RESET TO ORIGINAL DATA
Enter the option: 1

Your selection is SHOW THE DATA

| ID | Name | Director | Year | Dur | KPR | PLR | Watchdate |
|----|-----|-----|-----|-----|-----|-----|-----|
| 1 | Pulp Fiction | Quentin Tarantino | 1994 | 154 | 8.5 | 8.9 | 15.01.2021 |
| 4 | Princess Mononoke | Hayao Miyazaki | 1997 | 134 | 8.3 | 8.5 | 11.02.2024 |
| 5 | Schindler's List | Steven Spielberg | 1993 | 195 | 8.8 | 10.0 | 31.08.2020 |
| 6 | Whiplash | Damien Chazelle | 2014 | 107 | 8.4 | 10.0 | 18.01.2020 |
| 9 | Arrhythmia | Boris Khlebnikov | 2017 | 112 | 7.2 | 8.2 | 01.12.2023 |
| 10 | The Banshees of Inisherin | Martin McDonagh | 2022 | 114 | 7.5 | 6.4 | 05.01.2023 |
| 12 | Yellow Submarine | George Dunning | 1968 | 89 | 8.0 | 8.1 | 03.11.2023 |
| 13 | The Lighthouse | Robert Eggers | 2019 | 109 | 7.1 | 9.5 | 05.02.2024 |

Press ENTER to continue

```

Выводы.

В результате выполнения работы изучена и освоена работа с линейными односвязными списками.