**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МОЭВМ**

отчет

по лабораторной работе №4

по дисциплине «Программирование»

Тема: Линейные списки

Студентка гр. 6304 Курков Д. В.

Преподаватель Александр

Санкт-Петербург

2016

**Цель работы:** Сделать лабораторную работу**.**

**Ход работы.**

**#include <stdlib.h>**

#include <stdio.h>

#include <string.h>

#include <stddef.h>

//Структура MusicalComposition

**typedef** **struct** MusicalComposition

{

**char**\* name;

**char**\* author;

**int** year;

**struct** MusicalComposition \*next;

}MusicalComposition;

//Инициализация структуры

MusicalComposition\* createMusicalComposition (**char**\* name, **char**\* author, **int** year)

{

MusicalComposition\* composition = (MusicalComposition\*)malloc(**sizeof**(MusicalComposition));

composition->name = name;

composition->author = author;

composition->year = year;

composition->next = NULL;

**return** composition;

}

//Создание списка

MusicalComposition\* createMusicalCompositionList(**char**\*\* array\_names, **char**\*\* array\_authors, **int**\* array\_years, **int** n)

{

MusicalComposition\* head = createMusicalComposition(array\_names[0], array\_authors[0], array\_years[0]);

MusicalComposition\* tmp = head;

**for** (**int** i = 1; i < n; i++)

{

tmp->next = createMusicalComposition(array\_names[i], array\_authors[i], array\_years[i]);

tmp = tmp->next;

}

**return** head;

}

//Добавление нового элемента в список

**void** push (MusicalComposition\* head, MusicalComposition\* element)

{

MusicalComposition\* tmp = head;

**while** (tmp->next != NULL)

tmp = tmp->next;

tmp->next = element;

}

//Удаление элемента имеющего указанное имя из списка

**void** removeEl(MusicalComposition\* head, **char**\* name\_for\_remove)

{

MusicalComposition \*bef = head, \*tmp = head->next;

**while** (tmp != NULL)

{

**if** (strcmp(tmp->name, name\_for\_remove) == 0)

{

bef->next = tmp->next;

tmp->next;

}

bef = bef->next;

tmp = tmp->next;

}

}

//Счетчик элементов списка

**int** count (MusicalComposition\* head)

{

MusicalComposition\* tmp = head;

**int** count = 0;

**while** (tmp != NULL)

{

count++;

tmp = tmp->next;

}

**return** count;

}

//Вывод имен элементов списка

**void** print\_names(MusicalComposition \*head)

{

MusicalComposition \*tmp = head;

**while** (tmp != NULL)

{

printf ("%s\n", tmp->name);

tmp = tmp->next;

}

}

**int** main()

{

**int** length;

scanf("%d\n", &length);

**char**\*\* names = (**char**\*\*)malloc(**sizeof**(**char**\*)\*length);

**char**\*\* authors = (**char**\*\*)malloc(**sizeof**(**char**\*)\*length);

**int**\* years = (**int**\*)malloc(**sizeof**(**int**)\*length);

**for** (**int** i=0;i<length;i++)

{

**char** name[80];

**char** author[80];

fgets(name, 80, stdin);

fgets(author, 80, stdin);

fscanf(stdin, "%d\n", &years[i]);

(\*strstr(name,"\n"))=0;

(\*strstr(author,"\n"))=0;

names[i] = (**char**\*)malloc(**sizeof**(**char**\*) \* (strlen(name)+1));

authors[i] = (**char**\*)malloc(**sizeof**(**char**\*) \* (strlen(author)+1));

strcpy(names[i], name);

strcpy(authors[i], author);

}

MusicalComposition\* head = createMusicalCompositionList(names, authors, years, length);

**char** name\_for\_push[80];

**char** author\_for\_push[80];

**int** year\_for\_push;

**char** name\_for\_remove[80];

fgets(name\_for\_push, 80, stdin);

fgets(author\_for\_push, 80, stdin);

fscanf(stdin, "%d\n", &year\_for\_push);

(\*strstr(name\_for\_push,"\n"))=0;

(\*strstr(author\_for\_push,"\n"))=0;

MusicalComposition\* element\_for\_push = createMusicalComposition(name\_for\_push, author\_for\_push, year\_for\_push);

fgets(name\_for\_remove, 80, stdin);

(\*strstr(name\_for\_remove,"\n"))=0;

printf("%s %s %d\n", head->name, head->author, head->year);

**int** k = count(head);

printf("%d\n", k);

push(head, element\_for\_push);

k = count(head);

printf("%d\n", k);

removeEl(head, name\_for\_remove);

print\_names(head);

k = count(head);

printf("%d\n", k);

**return** 0;

}

**Вывод:** задание выполнено