

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

ОТЧЕТ
ПО КУРСОВОЙ РАБОТЕ
по дисциплине «Программирование»
ТЕМА: «Линейные списки»

Студенты гр. 7381

Судакова П. С.

Преподаватель

Берленко Т. А.

Санкт-Петербург

2017

Аннотация

Реализована функция работы со списком, которая добавляет в голову переданного списка вставляет 3 других списка. В ней используется функция `createMusicalComposition` из 4 лабораторной работы. Так же для демонстрации была разработана функция `print_name`, чтобы выводить список песен по переданному указателю на произвольный элемент списка.

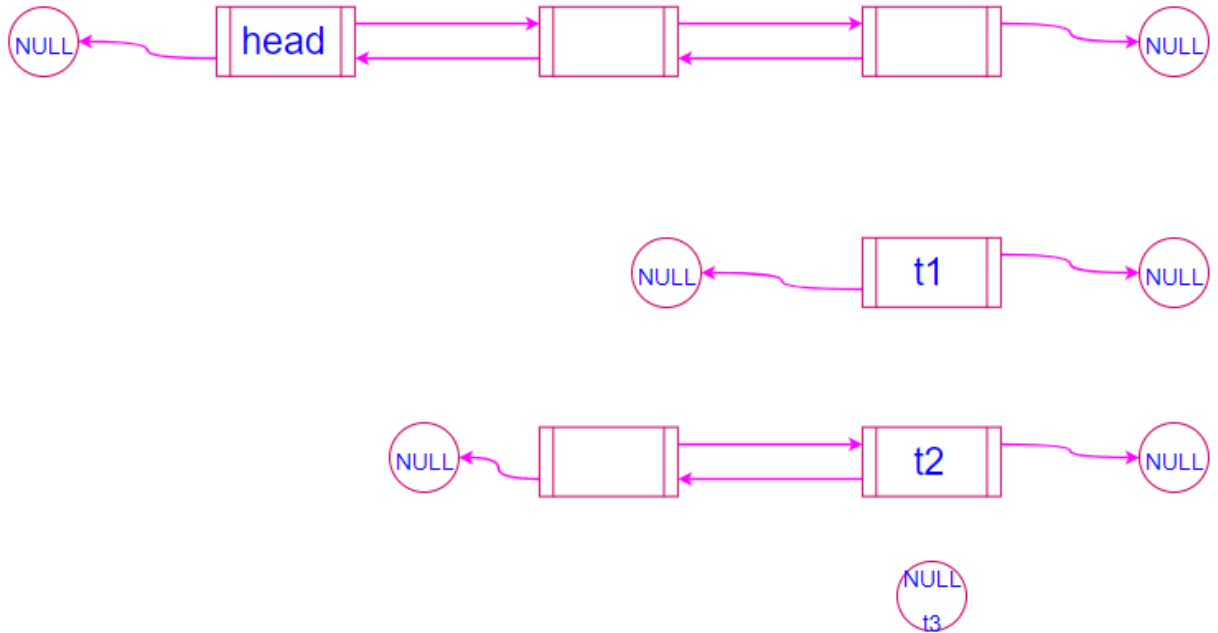
Цель: Реализовать функцию работы с двунаправленным списком, которая вставляет 3 произвольных элемента в список.

Задание: Добавить в начало списка 3 элемента (имя, автор, год - произвольные) на вход функции подаются 4 указателя.

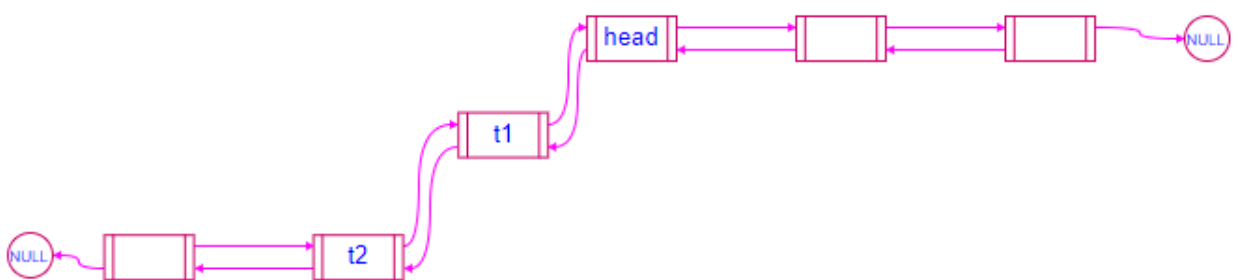
Задачи: Для достижения цели были поставлены следующие задачи

- Разработать визуальное представление данной задачи

Входные данные



Реализация



- Реализовать визуальное представление в рабочий код. Используется функция `createMusicalCompositin` из 4 лабораторной работы. Код представлен в Приложении А.
- Тестирование полученной программы.
Входные данные -4 списка. 1- список из 5 элементов; 2-список из одного элемента; 3-список из трех элементов, но указатель не на голову; 4 – указатель на NULL. Воспользуемся функцией

добавления в голову 1 списка всех остальных. И выведем этот список.

Входные данные, визуальное представление:

```
1
Rinat
Ctefaz Ctefaz
Lucia
Frida
Norman
2
fsahsd
3
Rinat
Ctefaz Ctefaz
Lucia
4
```

Выходные данные, визуальное представление:

```
2
Rinat
Ctefaz Ctefaz
Lucia
fsahsd
Rinat
Ctefaz Ctefaz
Lucia
Frida
Norman
```

Вывод:

В результате выполнения курсовой работы были на практике изучены и самостоятельно реализованы двунаправленные списки. Были учтены их слабые места и их плюсы.

Исходный код:

```
#include "metodFromLr4.h"
#include "string.h"
#include "stdlib.h"

MusicalComposition* createMusicalComposition(char *name, char* autor, int year){
    MusicalComposition *temp=malloc(1 * sizeof(MusicalComposition));
    temp->autor = autor;
    temp->name = name;
    temp->year = year;
    temp->next=NULL;
    temp->prev=NULL;
    return temp;
}

MusicalComposition* createMusicalCompositionList(char **arrayNames, char** arrayAutor, int*
arrayYear, int n){
    if (n >= 0){
        if (n==1) return createMusicalComposition(arrayNames[0], arrayAutor[0],
arrayYear[0]);
        MusicalComposition *head = createMusicalComposition(arrayNames[0],
arrayAutor[0], arrayYear[0]) ,
        *temp = createMusicalComposition(arrayNames[1], arrayAutor[1],
arrayYear[1]), *current = head, *prevCurent=NULL;
        for (int i = 1; i <= n; i++){
            current->prev = prevCurent;
            if (i == n) break;
            current->next = temp;
            prevCurent = current;
            current = temp;
            temp = createMusicalComposition(arrayNames[i+1], arrayAutor[i+1],
arrayYear[i+1]);
        }
        return head;
    }
    return NULL;
}

#include <stddef.h>
#include "courseWork.h"
void addHead(MusicalComposition *current, MusicalComposition *temp){
    if (current==NULL) {
        current = temp; return;
    }
    if (temp==NULL) {
        return;
    }
    for (; current->prev!=NULL; current = current->prev);
    for (; temp->next!=NULL; temp = temp->next);
    temp->next = current;
    current->prev = temp;
}

void addFromHeadAll4Element(MusicalComposition* head, MusicalComposition* t1,
MusicalComposition* t2, MusicalComposition* t3){
    addHead(head, t1);
    addHead(head, t2);
    addHead(head, t3);
}
```