

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

Кафедра вычислительной техники

**Отчет по лабораторной работе № 11-12**  
**по дисциплине «Программирование»**  
**Тема: Линейные двусвязные списки, кольцевые**  
**односвязные списки**

Студент гр. 9305

Салауров Е. М.

Преподаватель

Перязева Ю. В.

Санкт-Петербург

2020

## **Оглавление**

<b>Введение.....</b>	<b>3</b>
<b>Задание .....</b>	<b>3</b>
<b>Постановка задачи и описание решения .....</b>	<b>3</b>
<b>Описание переменных .....</b>	<b>5</b>
<b>Текст программы .....</b>	<b>12</b>
<b>Пример работы программы .....</b>	<b>13</b>
<b>Заключение .....</b>	<b>15</b>

## Введение

### Задание

Лабораторная работа № 11

Перепроектировать структуру, созданную при выполнении лабораторной работы №9 (по выбранной предметной области), так, чтобы одно из информационных полей, содержащих характеристику группы объектов (издательство, модель, тип и т. п.), стало ссылкой на элемент двусвязного линейного списка и выполнить задание в соответствии с вариантом.

Разработать подалгоритм и написать функцию, вставляющую в двусвязный список получаемые данные перед заданным по номеру элементом. Номер элемента задается с конца списка. При недостаточном количестве элементов в списке данные вставить в начало списка.

Лабораторная работа № 12

На основе линейного списка, созданного при выполнении лабораторной работы № 11 и содержащего одно информационное поле, выполнить задание в соответствии с вариантом.

Разработать подалгоритм вывода значений информационного поля элементов односвязного кольцевого списка в прямом или обратном направлении по желанию пользователя с одновременным удалением последнего выводимого на экран элемента.

### Постановка задачи и описание решения

Для решения этой задачи изначально нужно объявить структуру, я сделал это так:

```
typedef struct ZNAK {  
    char NAME[MAXLEN];  
    char NIK[MAXLEN];  
    int DATE[3];  
    int chislopole1;  
    int chislopole2;  
    float chislopole3;  
    float chislopole4;  
} group;
```

Структура была взята из лабораторной работы номер 10.

Также объявил структуру для списка:

```
struct node{  
    int id;  
    group *data;  
    struct node *prev;  
    struct node *next;  
} node;
```

Здесь data это поле, которое содержит информацию, которая описывается в структуре ZNAK. В лабораторной работе номер 12 отсутствует поле с указателем на предыдущий элемент списка.

Так же объявил структуру для головы списка:

```
typedef struct head{
    int N;
    struct node *first;
    struct node *last;
}head;
```

Начало работы у обеих программ одинаковое, программы запускают обработку (fill\_list(q);) файла формата .csv и выводят меню в котором предлагают возможные действия со списком (command\_selecting(q);).

Для лабораторной работы 11:

Программа предлагает пользователю добавить узел в список (enterFromKeyboard(q);), вывести список и выйти. В случае добавления узла программа предлагает несколько вариантов:

1. Добавить в начало (add\_first(q);)
2. Добавить в конец (add\_last(q);)
3. Добавить в определенное место (тут программа ведет также проверку на корректный ввод индекса узла, если такого индекса не существует, то программа добавляет узел на первое место) (insert\_after(p, q -> N);)

Далее программа запрашивает данные которые она должна вставить в добавляемый узел и происходит вставка.

Так же пользователь может выбрать вывод списка тогда ему предлагается меню из двух пунктов, где можно вывести список в стандартном порядке (output\_list(head \*q)) или в реверсе (output\_list\_reverse(head \*q)).

Для лабораторной работы 12:

Тут программа выводит сформированный список и предлагает пользователю вывести кольцевой список в стандартном порядке (output\_list(head \*q)) или в реверсе (output\_list\_reverse(head \*q)) при этом происходит удаление последнего выводимого узла.

## Описание структуры:

Структура ZNAK:

Название переменной	Тип переменной	Назначение
NAME	char	Поле с именем
NIK	char	Поле со знаком зоди
DATE	int	Массив содержащий дату рождения
chislopole1	int	Любимое число пользователя
Chislopole2	int	Счастливое число пользователя
Chislopole3	float	Процент удачи пользователя
Chislopole4	float	

Структура node:

Название переменной	Тип переменной	Назначение
data	group	Информационное поле
id	int	Переменная номер структуры
prev	node	Указатель на предыдущую структуру в списке(Этого поля нет в кольцевом списке)
next	node	Указатель на следующую структуру в списке

Структура head:

Название переменной	Тип переменной	Назначение
N	int	Переменная счетчик структур
first	node	Указатель на первую структуру в списке
last	node	Указатель на последнюю структуру в списке

## Описание функций:

### Функция `main()`:

#### Описание:

Точка входа в программу. Отвечает за открытие файла, содержащего данные для последующей работы.

#### Прототип:

```
int main()
```

#### Пример вызова:

```
main()
```

#### Описание переменных:

Название переменной	Тип переменной	Назначение
q	head	Указатель на голову списка

**Возвращает значение:** 0, если работа программы завершена успешно.

### Функция `fill_list`

#### Описание:

Считывание информации из файла и записывания его в массив структуры. Пока строка не совпадет с предыдущей, программа ее разделяет и записывает в поля списка.

#### Прототип:

```
void fill_list(head *q)
```

#### Примеры вызова:

```
fill_list(q)
```

#### Описание переменных:

Название переменной	Тип переменной	Назначение
q	head	Указатель на голову списка
str	char	Массив для копирования
message	char	Массив для копирования

## Функция enterFromKeyboard

### Описание:

Ввод информации в базу данных с клавиатуры: в начало списка или в конец.

### Прототип:

```
void enterFromKeyboard(head *q)
```

### Примеры вызова:

```
enterFromKeyboard(q)
```

### Описание переменных:

Название переменной	Тип переменной	Назначение
q	head	Указатель на голову списка
k	int	Переменная для определения действия
temp	node	Узел списка

## Функция split():

### Описание:

Функция разделения строки по заданному разделителю.

Каждая строка файла разделяется на элементы промежуточного массива строк s по разделителям с помощью функции и в зависимости от типа поля элемента массива структур выполняется преобразование элемента массива строк в поле отдельной структуры.

### Прототип:

```
void split(char *mes, head *q)
```

### Пример вызова:

```
split(char *mes, head *q);
```

### Описание переменных:

Название переменной	Тип переменной	Назначение
temp	node	Указатель на структуру
s	char	Массив для копирования данных
n	int	Переменная для корректной записи данных в структуру
k	int	длина выделяемой подстроки

**Возвращаемое значение:** массив строк.

### Функция `str_len ()`:

#### Описание:

длина строки.

#### Прототип:

```
int str_len(char *s)
```

#### Пример вызова:

```
str_len(s)
```

#### Описание переменных:

Название переменной	Тип переменной	Назначение
s	char	строка
r	int	количество символов

### Функция `command_selecting ()`:

#### Описание:

работа с пользователем, то есть программа не завершается пока пользователь не захочет этого (введет нужную команду), то есть для добавления элемента - 1, для удаления - 2, для вывода списка - 3, выхода из программы 4. То есть работает по не введено 4.

#### Прототип:

```
void command_selecting(head *q)
```

#### Пример вызова:

```
command_selecting(q)
```

#### Описание переменных:

Название переменной	Тип переменной	Назначение
q	head	голова списка
s	char	команда
f	int	индификатор команды

### Функция `free_head()`:

#### Описание:

очищение головы списка.

#### Прототип:

```
void free_head(head *q)
```

#### Примеры вызова:

```
free_head(q)
```

#### Описание переменных:

Название переменной	Тип переменной	Назначение
q	head	голова списка

### Функция `free_node ()`:

#### Описание:

очищение узла.

#### Прототип:

```
void free_node(node *temp)
```

#### Примеры вызова:

```
free_node(temp)
```

#### Описание переменных:

Название переменной	Тип переменной	Назначение
q	head	голова списка



### Функция free\_list ():

#### Описание:

очищение всего списка, пока у последнего узла не будет ссылка на NULL.

#### Прототип:

```
void free_list(head *q)
```

#### Примеры вызова:

```
free_list(q)
```

#### Описание переменных:

Название переменной	Тип переменной	Назначение
p	node	предыдущий узел
temp	node	узел
q	head	голова списка

### Функция malloc\_node ():

#### Описание:

выделяет память полям узла.

#### Прототип:

```
void malloc_node(node *temp)
```

#### Примеры вызова:

```
malloc_node(temp)
```

#### Описание переменных:

Название переменной	Тип переменной	Назначение
temp	node	узел

### Функция create\_head ():

#### Описание:

создание головы списка.

#### Прототип:

```
head *create_head()
```

#### Примеры вызова:

```
create_head()
```

#### Описание переменных:

Название переменной	Тип переменной	Назначение
p	head	голова списка

### Функция add\_last ():

#### Описание:

добавление узла в конец списка.

#### Прототип:

```
void add_last(head *q)
```

#### Примеры вызова:

```
add_last(q)
```

#### Описание переменных:

Название переменной	Тип переменной	Назначение
temp	node	узел
q	head	голова списка

### Функция add\_first ():

#### Описание:

добавление узла в начало списка

#### Прототип:

```
node *add_first(head *q)
```

#### Примеры вызова:

```
add_first(q)
```

#### Описание переменных:

Название переменной	Тип переменной	Назначение
temp	node	узел
q	head	голова списка

### Функция delete\_first ():

#### Описание:

удаление первого узла

#### Прототип:

```
void delete_first(head *q)
```

#### Примеры вызова:

```
delete_first(q)
```

#### Описание переменных:

Название переменной	Тип переменной	Назначение
temp	node	узел
q	head	голова списка

### Функция delete\_node ():

#### Описание:

удаление любого узла, кроме первого.

#### Прототип:

```
void delete_node(node *p, head *q)
```

#### Примеры вызова:

```
delete_node(p, q)
```

#### Описание переменных:

Название переменной	Тип переменной	Назначение
temp	node	узел
q	head	голова списка
p	node	предыдущий узел

### Функция output\_list\_reverse ():

#### Описание:

Печать базы данных от последней записи к первой.

#### Прототип:

```
void output_list_reverse(head *q)
```

#### Пример вызова:

```
output_list_reverse(q)
```

#### Описание переменных:

Название переменной	Тип переменной	Назначение
temp	node	узел
q	head	голова списка

### Функция create\_node():

#### Описание:

Создание узла в списке

#### Прототип:

```
node *create_node (head *q)
```

#### Пример вызова:

```
create_node(head *q);
```

#### Описание переменных:

Название переменной	Тип переменной	Назначение
temp	node	Структура для для создания списка

### Функция struct\_fill():

#### Описание:

Функция заполнения структуры данными из файла. В массив строк вводятся полученные из simple\_split данные.

#### Прототип:

```
cars *struct_fill(char **str)
```

#### Пример вызова:

```
ch[i]=struct_fill(s2);
```

#### Описание переменных:

Название переменной	Тип переменной	Назначение
str0	group	Структура для распределения в ней данных из массива

### Функция print\_list():

#### Описание:

Функция вывода списка. Пока указатель на голову не NULL, выводит данные одной строки и переходит к следующей

#### Прототип:

```
void print_list(node *head)
```

#### Пример вызова:

```
print_list(head);
```

#### Описание переменных:

Название переменной	Тип переменной	Назначение
p	node	Указатель на голову списка

## Функция insert\_after():

### Описание:

Вставляет новый узел в список при этом учитывает то, что пользователь вводит несуществующий индекс в списке.

### Прототип:

```
void insert_after(node *value, int index)
```

### Пример вызова:

```
insert_after_DLL(node *lst, int N)
```

### Описание переменных:

Название переменной	Тип переменной	Назначение
N	int	Переменная для ориентирования в списке
p	node	Указатель на данные из головы массива
temp	node	Структура которую вставляют в список

## Текст программы

Для лабораторной работы 11:

[https://github.com/ZZANZZAN/LAB\\_PROG/tree/master/LAB11-12/LAB11](https://github.com/ZZANZZAN/LAB_PROG/tree/master/LAB11-12/LAB11)

Для лабораторной работы 12:

[https://github.com/ZZANZZAN/LAB\\_PROG/tree/master/LAB11-12/LAB12](https://github.com/ZZANZZAN/LAB_PROG/tree/master/LAB11-12/LAB12)

# Пример работы программы

## Лабораторная работа 11:

```
D:\-еэю\Programming\LAB11\bin\Debug\LAB11.exe
| 9|      Stone Sharonne |      riba|2001| 5| 3|  234|      3|  0.550000|  0.354000|
| | Menu: |
+-+-----+
|1| - Add stuct |
|2| - Print struct|
|0| - Exit |
Your choice:
1
| | Add: |
+-+-----+
|1| - Add first |
|2| - Add end |
|3| - Enter index |
Your choice:
2
Enter name: 1
Enter Zodiac sign: 1
Enter year: 1
Enter mounth: 1
Enter day: 1
Enter like chislo:
1
Enter good chislo:
1
Enter percentage of luck:
1
Enter confidence percentage:
1
| | Menu: |
+-+-----+
|1| - Add stuct |
|2| - Print struct|
|0| - Exit |
2
| | Select the printing method: |
+-+-----+
|1| - At first |
|2| - From the end(do not work in this version)|
Command:
1
Initial array:
| N|      NAME|      Nik|      Date| int1| int2|      float1|      float2|
+-+-----+
| 1|      Last James |      vodolei|2000| 3| 2|      1|      12|  0.120000|  0.234000|
| 2|      Mae Vanessa |      lev|1990| 2| 1|     123|     123|  0.220000|  0.124000|
| 3|      Chang Jackie |      oven|1999| 1| 4|     213|     34|  0.110000|  0.414000|
| 4|      Stone Sharonne |      riba|2001| 5| 3|     234|      3|  0.550000|  0.354000|
| 5|      McCartney Pol |      ckorpion|1999| 8| 2|      3|      1|  0.880000|  0.284000|
| 6|      Howston Witney |      vodolei|1987| 6| 1|      1|      3|  0.660000|  0.164000|
| 7|      Stone Sharonne |      riba|2001| 5| 3|     13|      2|  0.550000|  0.354000|
| 8|      McCartney Pol |      lev|1990| 2| 1|     23|     342|  0.220000|  0.124000|
| 9|      Stone Sharonne |      riba|2001| 5| 3|     234|      3|  0.550000|  0.354000|
|10|      1 |      1|  1| 1| 1|      1|      1|  1.000000|  1.000000|
| | Menu: |
+-+-----+
|1| - Add stuct |
|2| - Print struct|
|0| - Exit |
```

## Лабораторная работа 12:

```
D:\-еэзю\Programming\LAB12\bin\Debug\LAB12.exe
| 2|      Mae Vanessa |      lev|1990| 2| 1| 123| 123| 0.220000| 0.124000|
| 3|      Chang Jackie|      oven|1999| 1| 4| 213| 34| 0.110000| 0.414000|
| 4|      Stone Sharonne|      riba|2001| 5| 3| 234| 3| 0.550000| 0.354000|
| 5|      McCartney Pol|      ckorpion|1999| 8| 2| 3| 1| 0.880000| 0.284000|
| 6|      Howston Witney|      vodolei|1987| 6| 1| 1| 3| 0.660000| 0.164000|
| 7|      Stone Sharonne|      riba|2001| 5| 3| 13| 2| 0.550000| 0.354000|
| 8|      McCartney Pol|      lev|1990| 2| 1| 23| 342| 0.220000| 0.124000|
| 9|      Stone Sharonne|      riba|2001| 5| 3| 234| 3| 0.550000| 0.354000|

| | Menu: |
+-----+
|1| - Add stuct(do not work in this version)|
|2| - Print struct
|0| - Exit
Your choice:
2
| | Select the printing method: |
+-----+
|1| - At first
|2| - From the end(do not work in this version)|
Command:
1
Initial array:
| N|      NAME|      Nik|      Date| int1| int2|      float1|      float2|
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1|      Last James|      vodolei|2000| 3| 2| 1| 12| 0.120000| 0.234000|
| 2|      Mae Vanessa|      lev|1990| 2| 1| 123| 123| 0.220000| 0.124000|
| 3|      Chang Jackie|      oven|1999| 1| 4| 213| 34| 0.110000| 0.414000|
| 4|      Stone Sharonne|      riba|2001| 5| 3| 234| 3| 0.550000| 0.354000|
| 5|      McCartney Pol|      ckorpion|1999| 8| 2| 3| 1| 0.880000| 0.284000|
| 6|      Howston Witney|      vodolei|1987| 6| 1| 1| 3| 0.660000| 0.164000|
| 7|      Stone Sharonne|      riba|2001| 5| 3| 13| 2| 0.550000| 0.354000|
| 8|      McCartney Pol|      lev|1990| 2| 1| 23| 342| 0.220000| 0.124000|

| | Menu: |
+-----+
|1| - Add stuct(do not work in this version)|
|2| - Print struct
|0| - Exit
2
| | Select the printing method: |
+-----+
|1| - At first
|2| - From the end(do not work in this version)|
Command:
2
Initial array:
| N|      NAME|      Nik|      Date| int1| int2|      float1|      float2|
+-----+-----+-----+-----+-----+-----+-----+
| 7|      Stone Sharonne|      riba|2001| 5| 3| 13| 2| 0.550000| 0.354000|
| 6|      Howston Witney|      vodolei|1987| 6| 1| 1| 3| 0.660000| 0.164000|
| 5|      McCartney Pol|      ckorpion|1999| 8| 2| 3| 1| 0.880000| 0.284000|
| 4|      Stone Sharonne|      riba|2001| 5| 3| 234| 3| 0.550000| 0.354000|
| 3|      Chang Jackie|      oven|1999| 1| 4| 213| 34| 0.110000| 0.414000|
| 2|      Mae Vanessa|      lev|1990| 2| 1| 123| 123| 0.220000| 0.124000|
| 1|      Last James|      vodolei|2000| 3| 2| 1| 12| 0.120000| 0.234000|

| | Menu: |
+-----+
|1| - Add stuct(do not work in this version)|
|2| - Print struct
|0| - Exit
```

## **Заключение**

### **Выводы:**

При выполнении лабораторной работы были получены практические навыки в разработке алгоритма и написании программы на языке Си. Были получены основные знания о синтаксисе языка Си, в частности, о программировании задач со структурами, а также правилах написания кода на языке Си.