

Лабораторная работа 4.1 «Динамические структуры данных»

1. Двусвязный список

Двусвязный список позволяет двигаться по цепочке элементов в обоих направлениях, имея доступными следующий и предыдущий элементы.

Формирование двусвязного списка:

```
struct List
{
    int info;
    struct List *next;
    struct List *prev;
};
```

Циклический список позволяет моделировать линейные цепочки элементов, исключив постоянные проверки на «первый» и «последний». В таком списке поле `next` последнего элемента ссылается на первый элемент, а поле `prev` первого - на последний элемент списка. Единственный элемент списка ссылается сам на себя (`l->next = l; l->prev = l`). Операции включения элемента в начало и конец списка идентичны за исключением того, что в первом случае меняется указатель на первый элемент списка.

На рисунках 1, 2 приведены примеры двусвязного и циклического списков.

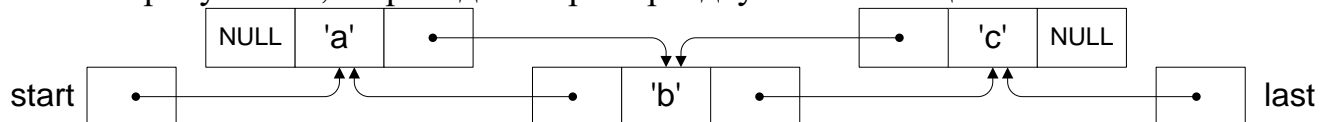


Рисунок 1 – Двусвязный список

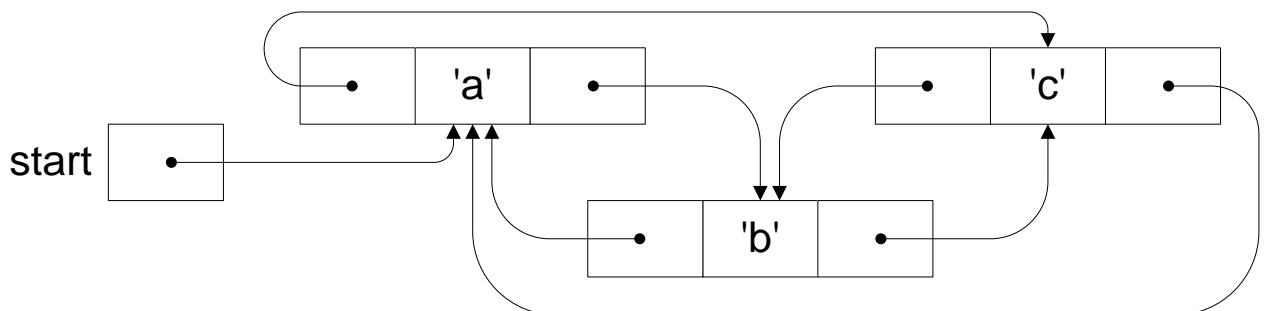


Рисунок 2 – Циклический двусвязный список

1.1 Добавление элементов в двусвязный список

Определена структура, которая будет использоваться в последующих примерах:

```
#define LIST struct List

LIST
{
    char info;
    LIST *next;
    LIST *prev;
};
```

Функция добавления элементов в двусвязный список:

```
void add(LIST **current, char item)    //current - указатель на текущий элемент
{                                       списка
```

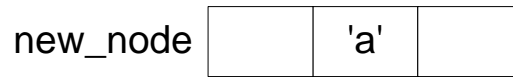
LIST *new_node;	//новый элемент двусвязного списка
new_node = new LIST;	//создаем новый элемент двусвязного списка
new_node->info = item;	//заполняем поле info
if (*current == NULL)	//добавление первого элемента двусвязного списка
{	
*current=new_node;	//текущим элементом становится новый элемент
new_node->next=NULL;	//указатель на следующий элемент после нового равен нулю (т.е. элемент не существует)
new_node->prev=NULL;	//указатель на предыдущий элемент по отношению к новому равен нулю (т.е. элемент не существует)
}	
if ((*current)->next!=NULL)	//добавление элемента в середину двусвязного списка (после текущего элемента)
{	
new_node->next=(*current)->next;	//следующим по отношению к новому элементу становится элемент списка следующий после текущего
(*current)->next->prev=new_node;	//предыдущим по отношению к элементу следующего после текущего становится новый элемент
new_node->prev=*current;	//предыдущим по отношению к новому становится текущий элемент
(*current)->next=new_node;	//следующим по отношению к текущему элементу становится новый элемент
}	
else	//добавление элемента в конец двусвязного списка
{	
(*current)->next=new_node;	//следующим по отношению к текущему становится новый элемент
new_node->prev=*current;	//предыдущим по отношению к новому становится текущий элемент списка
new_node->next=NULL;	//указатель на следующий элемент после нового равен нулю (т.е. элемента не существует)
(*current)=new_node;	//текущим элементом становится новый элемент списка
}	

Пример. Пусть необходимо сформировать двусвязный список, представленный в программе переменной *l*, состоящий из трех элементов ('a','c','b') (элементы указаны в порядке их добавления в список), упорядоченный по алфавиту. Для этого необходимо использовать функцию add:

```
LIST *l = 0;
add(&l, 'a');
add(&l, 'c');
l = l->prev;
add(&l, 'b');
```

На рис. 3, 4, 5 показаны примеры добавления элементов в начало, середину и конец двусвязного списка соответственно.

```
new_node = new LIST; new_node->info = item;
```

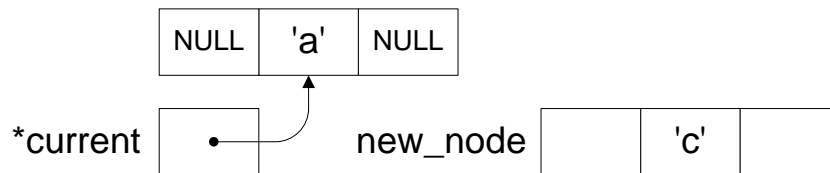


```
If(*current==NULL){*current = new_node;  
new_node->next = NULL;  
new_node->prev = NULL;}
```

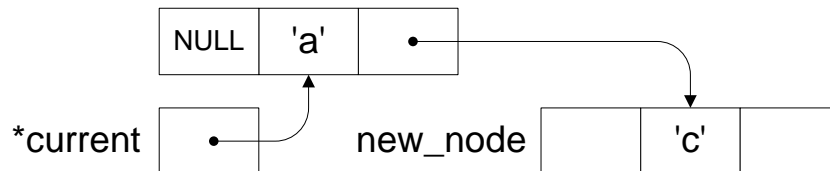


Рисунок 3 – Добавление первого элемента двусвязного списка

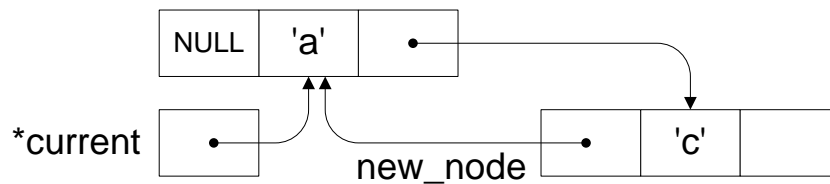
```
new_node = new LIST; new_node->info = item;
```



```
(*current)->next = new_node;
```



```
new_node->prev = *current
```



```
new_node->next = NULL; (*current) = new_node;
```

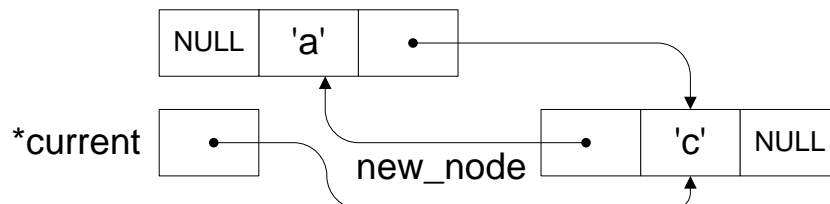


Рисунок 4 – Добавление элемента в конец двусвязного списка

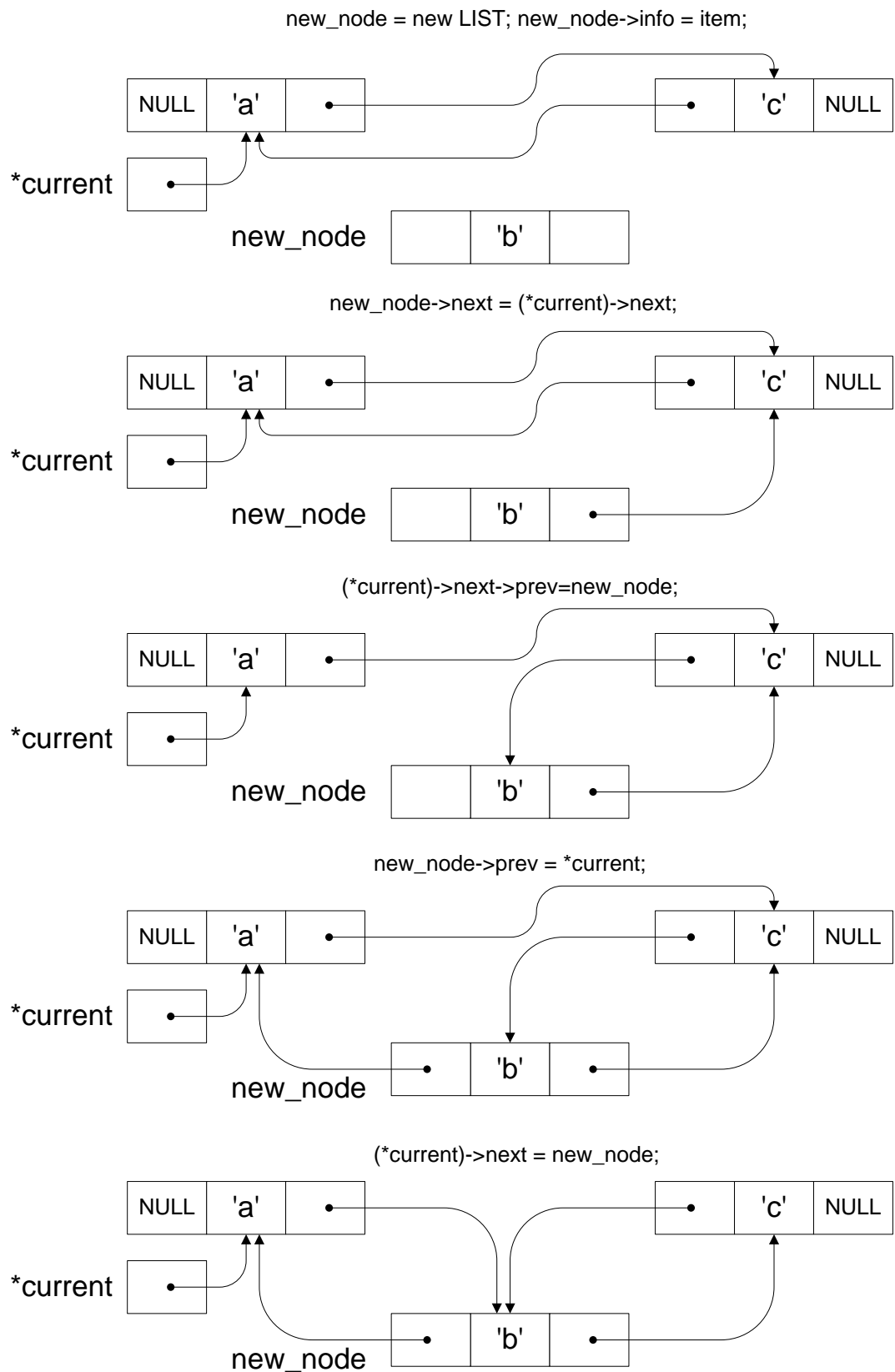


Рисунок 5 – Добавление элемента в середину двусвязного списка

1.2 Удаление элементов из двусвязного списка

Функция удаления элементов из двусвязного списка:

<pre>void del(LIST **current) { LIST *old_node=*current; if ((*current)->prev==NULL)</pre>	<pre>//current - указатель на текущий элемент //old_node - указатель на удаляемый //элемент списка //удаление первого элемента списка</pre>
---	---

<pre> { *current=(*current)->next; if (*current) (*current)->prev=NULL; } else { if ((*current)->next==NULL) { (*current)=(*current)->prev; (*current)->next=NULL; } else { (*current)->prev->next=(*current)->next; (*current)->next->prev=(*current)->prev; (*current)=(*current)->next; } } free(old_node); } </pre>	<pre> //текущим элементом становится следующий элемент списка //если следующий элемент существует (в списке больше одного элемента) //указатель на предыдущий элемент по отношению к текущему равен нулю //удаление последнего элемента списка //текущим элементом становится предыдущий элемент списка //указатель на следующий элемент равен 0 //удаление элемента из середины списка //следующим элементом за предыдущим по отношению к текущему элементу становится следующий за текущим элемент //предыдущим элементом по отношению к следующему за текущим элементом становится предыдущий по отношению к текущему элемент //указатель current перемещается на следующий элемент списка //освобождение памяти, занимаемой текущим элементом списка </pre>
--	--

Пример. Пусть из двусвязного списка ('a','b','c'), представленного в программе переменной *l* необходимо удалить все элементы в следующей последовательности 'b', 'c', 'a'. Предполагается, что указатель *l* указывает на первый элемент двусвязного списка ('a'). Для этого необходимо использовать функцию `del` следующим образом:

```

l = l->next;
del(&l);
del(&l);
del(&l);

```

На рис. 6, 7, 8 показаны примеры удаления первого элемента двусвязного списка, удаление элемента из середины двусвязного списка и удаление последнего элемента двусвязного списка.

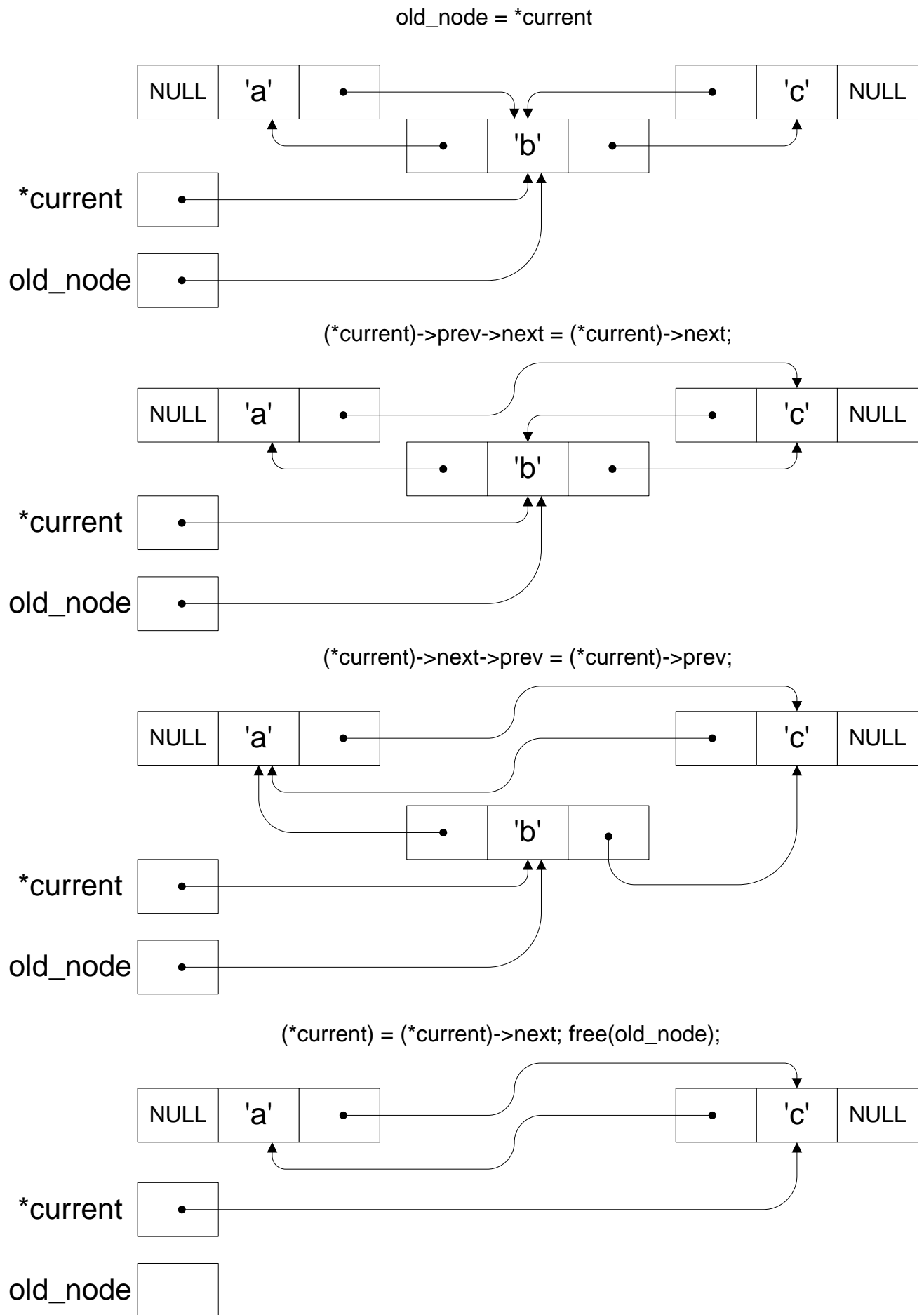


Рис. 6 – Удаление элемента из середины двусвязного списка

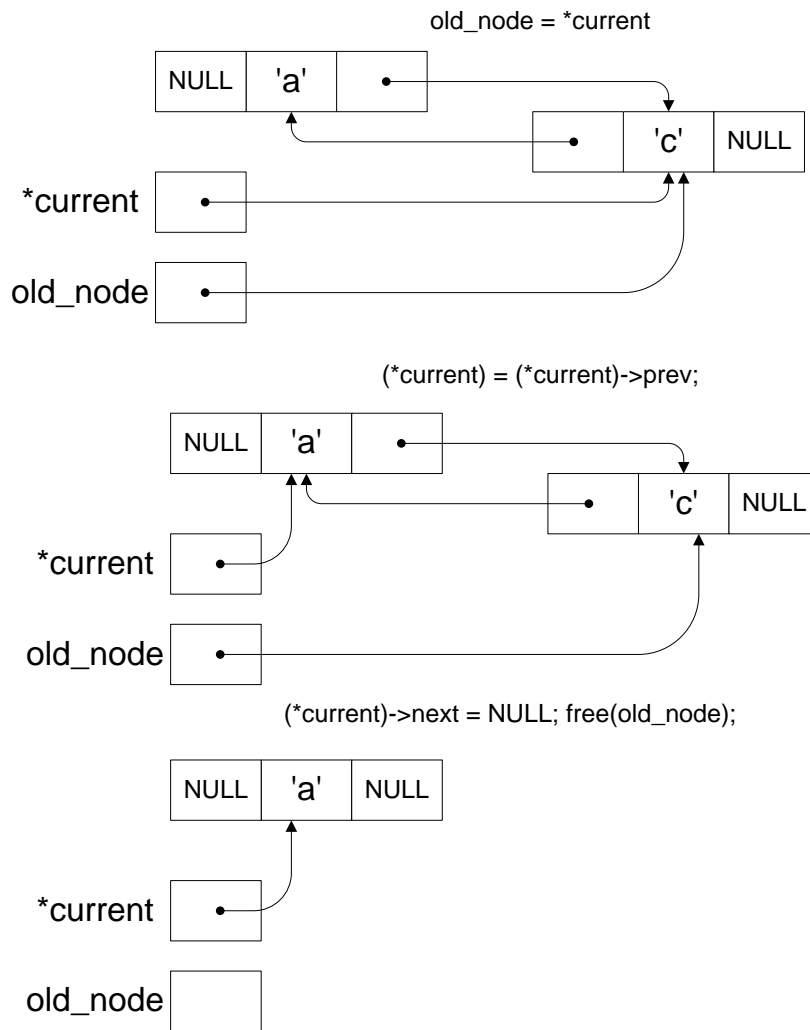


Рисунок 7 – Удаление последнего элемента двусвязного списка

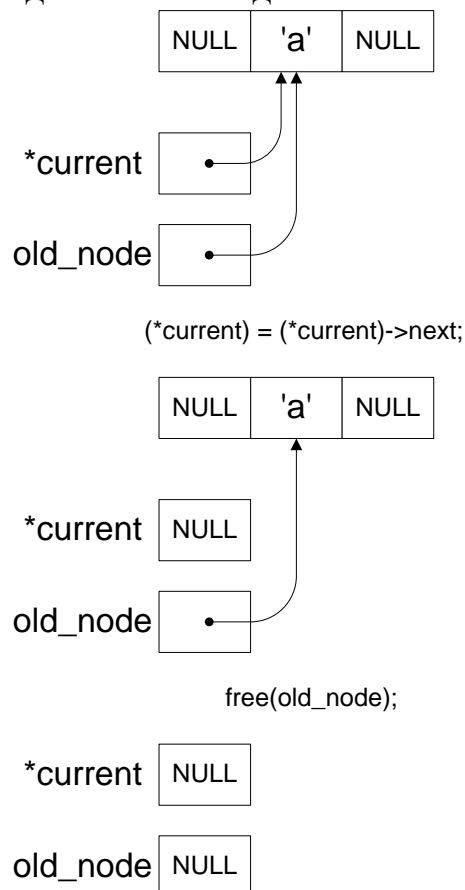


Рисунок 8 – Удаление первого элемента двусвязного списка

2. Варианты заданий к лабораторной работе 4.1

Вариант 1

1.) Составить программу, которая содержит текущую информацию об успеваемости студентов.

Сведения о студентах включают:

- фамилия и инициалы;
- номер группы;
- успеваемость (массив из пяти элементов).

2.) Программа должна обеспечивать:

- хранение сведений обо всех студентах в виде двусвязного списка, записи должны быть упорядочены по возрастанию номера группы;
- добавление данных о новых студентах;
- удаление данных о студенте, фамилия которого введена с клавиатуры;
- вывод сведений обо всех студентах;
- по запросу выводятся сведения о студентах, средний балл которых больше 4.0.

3.) Программа должна обеспечивать диалог с помощью меню.

Вариант 2

1.) Составить программу, которая содержит текущую информацию об успеваемости студентов.

Сведения о студентах включают:

- фамилия и инициалы;
- номер группы;
- успеваемость (массив из пяти элементов).

2.) Программа должна обеспечивать:

- хранение сведений обо всех студентах в виде двусвязного списка, записи должны быть упорядочены по возрастанию среднего балла;
- добавление данных о новых студентах;
- удаление данных о студенте, фамилия которого введена с клавиатуры;
- вывод сведений обо всех студентах;
- по запросу выводятся сведения о студентах, которые имеют оценки 4 и 5.

3.) Программа должна обеспечивать диалог с помощью меню.

Вариант 3

1.) Составить программу, которая содержит текущую информацию об успеваемости студентов.

Сведения о студентах включают:

- фамилия и инициалы;
- номер группы;
- успеваемость (массив из пяти элементов).

2.) Программа должна обеспечивать:

- хранение сведений обо всех студентах в виде двусвязного списка, записи должны быть упорядочены по алфавиту;
- добавление данных о новых студентах;

- удаление данных о студенте, фамилия которого введена с клавиатуры;
- вывод сведений обо всех студентах;
- по запросу выводятся сведения о студентах, которые имеют хотя бы одну оценку 2.

3.) Программа должна обеспечивать диалог с помощью меню.

Вариант 4

1.) Составить программу, которая содержит текущую информацию о заявках на авиабилеты.

Каждая заявка включает:

- пункт назначения;
- номер рейса;
- фамилию и инициалы пассажира;
- желаемую дату вылета.

2.) Программа должна обеспечивать:

• хранение всех заявок в виде двусвязного списка, записи должны быть упорядочены по возрастанию номера рейса;

- добавление заявок в список;
- удаление заявки пассажира, фамилия которого введена с клавиатуры;
- вывод всех заявок;
- вывод заявок по заданному пункту назначения.

3.) Программа должна обеспечивать диалог с помощью меню.

Вариант 5

1.) Составить программу, которая содержит текущую информацию о заявках на авиабилеты.

Каждая заявка включает:

- пункт назначения;
- номер рейса;
- фамилию пассажира;
- желаемую дату вылета.

2.) Программа должна обеспечивать:

• хранение всех заявок в виде двусвязного списка, записи должны быть размещены в алфавитном порядке по названиям пунктов назначения;

- добавление заявок в список;
- удаление заявки пассажира, фамилия которого введена с клавиатуры;
- вывод всех заявок;
- вывод заявок по заданной дате вылета.

3.) Программа должна обеспечивать диалог с помощью меню.

Вариант 6

1.) Составить программу, которая содержит текущую информацию о работниках организации.

Сведения о работнике включают:

- фамилия и инициалы работника;
- название занимаемой должности;

- год поступления на работу;
- 2.) Программа должна обеспечивать:
 - хранение сведений обо всех работниках в виде двусвязного списка, записи должны быть размещены по алфавиту;
 - добавление новых работников в список;
 - удаление работника из списка, фамилия которого введена с клавиатуры;
 - вывод сведений обо всех работниках;
 - по запросу выводятся сведения о работниках, чей стаж работы в организации превышает значение, введенное с клавиатуры.
- 3.) Программа должна обеспечивать диалог с помощью меню.

Вариант 7

- 1.) Автоматизированная информационная система на железнодорожном вокзале содержит сведения об отправлении поездов дальнего следования. Для каждого поезда указывается:
 - пункт назначения;
 - номер поезда;
 - время отправления.
- 2.) Программа должна обеспечивать:
 - хранение данных в информационной системе в виде двусвязного списка, записи должны быть размещены в алфавитном порядке по названиям пунктов назначения;
 - добавление данных о поездах в информационную систему;
 - удаление данных о поезде по введенному номеру поезда;
 - вывод информации обо всех поездах;
 - вывод информации о поездах, отправляющихся после введенного с клавиатуры времени.
- 3.) Программа должна обеспечивать диалог с помощью меню.

Вариант 8

- 1.) Автоматизированная информационная система на железнодорожном вокзале содержит сведения об отправлении поездов дальнего следования. Для каждого поезда указывается:
 - пункт назначения;
 - номер поезда;
 - время отправления.
- 2.) Программа должна обеспечивать:
 - хранение данных в информационной системе в виде двусвязного списка, записи должны быть упорядочены по времени отправления поезда;
 - добавление данных о поездах в информационную систему;
 - удаление данных о поезде по введенному номеру поезда;
 - вывод информации обо всех поездах;
 - вывод информации обо всех поездах, следующих до заданного пункта назначения.
- 3.) Программа должна обеспечивать диалог с помощью меню.

Вариант 9

1.) Автоматизированная информационная система на железнодорожном вокзале содержит сведения об отправлении поездов дальнего следования.

Для каждого поезда указывается:

- пункт назначения;
- номер поезда;
- время отправления.

2.) Программа должна обеспечивать:

• хранение данных в информационной системе в виде двусвязного списка, записи должны быть упорядочены по номерам поездов;

- добавление данных о поездах в информационную систему;
- удаление данных о поезде по введенному номеру поезда;
- вывод информации обо всех поездах;
- вывод информации о поезде, номер которого введен с клавиатуры;

3.) Программа должна обеспечивать диалог с помощью меню.

Вариант 10

1.) Составить программу, которая содержит информацию о наличии автобусов в автобусном парке.

Сведения о каждом автобусе включают:

- номер автобуса;
- фамилию и инициалы водителя;
- номер маршрута.

2.) Программа должна обеспечивать:

• хранение данных обо всех автобусах в парке в виде двусвязного списка, записи должны быть расположены в алфавитном порядке по фамилии водителя;

• добавление данных об автобусах в список при въезде каждого автобуса в парк;

• удаление данных об автобусе из списка при выезде из парка, номер которого введен с клавиатуры;

• вывод информации обо всех автобусах, находящихся в парке;

• по запросу выводится информация об автобусе по введенной фамилии водителя.

3.) Программа должна обеспечивать диалог с помощью меню.

Вариант 11

1.) Составить программу, которая содержит информацию о наличии автобусов в автобусном парке.

Сведения о каждом автобусе включают:

- номер автобуса;
- фамилию и инициалы водителя;
- номер маршрута.

2.) Программа должна обеспечивать:

• хранение данных обо всех автобусах в парке в виде двусвязного списка, записи должны быть упорядочены по номеру маршрута;

- добавление данных об автобусах в список при въезде каждого автобуса в парк;
- удаление данных об автобусе из списка при выезде из парка, фамилия водителя которого введена с клавиатуры;
- вывод информации обо всех автобусах, находящихся в парке;
- по запросу выводится информация обо всех автобусах, работающих на маршруте, номер которого введен с клавиатуры.

3.) Программа должна обеспечивать диалог с помощью меню.

Вариант 12

1.) Составить программу, которая содержит сведения телефонной книжки.

Каждая запись включает:

- фамилия, имя;
- номер телефона;
- день рождения (массив из трех чисел).

2.) Программа должна обеспечивать:

- хранение всех записей в виде двусвязного списка, записи должны быть упорядочены по датам дней рождения;
- добавление новой записи;
- удаление из списка информации о человеке, фамилия которого введена с клавиатуры;
- вывод информации обо всех номерах телефонов;
- по запросу выводится информация о человеке, номер телефона которого введен с клавиатуры.

3.) Программа должна обеспечивать диалог с помощью меню.

Вариант 13

1.) Составить программу, которая содержит сведения телефонной книжки.

Каждая запись включает:

- фамилия, имя;
- номер телефона;
- день рождения (массив из трех чисел).

2.) Программа должна обеспечивать:

- хранение всех записей в виде двусвязного списка, записи должны быть размещены по алфавиту;
- добавление новой записи;
- удаление из списка информации о человеке, фамилия которого введена с клавиатуры;
- вывод информации обо всех номерах телефонов;
- по запросу выводится информация о людях, чьи дни рождения приходятся на месяц, значение которого введено с клавиатуры.

3.) Программа должна обеспечивать диалог с помощью меню.

Вариант 14

1.) Составить программу, которая содержит сведения телефонной книжки.

Каждая запись включает:

- фамилия, имя;
- номер телефона;
- день рождения (массив из трех чисел).

2.) Программа должна обеспечивать:

- хранение всех записей в виде двусвязного списка, записи должны быть упорядочены по трем первым цифрам номера телефона;
- добавление новой записи;
- удаление из списка информации о человеке, день рождения которого приходится на дату, введенную с клавиатуры;
- вывод информации обо всех номерах телефонов;
- по запросу выводится информация о человеке, чья фамилия введена с клавиатуры.

3.) Программа должна обеспечивать диалог с помощью меню.

Вариант 15

1.) Составить программу, которая содержит текущую информацию о книгах в библиотеке.

Сведения о книгах включают:

- фамилию и инициалы автора;
- название;
- год издания;
- количество экземпляров данной книги в библиотеке.

2.) Программа должна обеспечивать:

- хранение всех данных обо всех книгах в библиотеке в виде двусвязного списка, записи должны быть упорядочены по годам издания;
- добавление данных о книгах вновь поступивших в библиотеку;
- удаление данных о списываемой книге, название которой введено с клавиатуры;
- вывод информации обо всех книгах в библиотеке;
- по запросу выводится информация обо всех книгах автора, имеющихся в библиотеке, чья фамилия введена с клавиатуры.

3.) Программа должна обеспечивать диалог с помощью меню.

Вариант 16

1.) Составить программу, которая содержит текущую информацию о книгах в библиотеке.

Сведения о книгах включают:

- фамилию и инициалы автора;
- название;
- год издания;
- количество экземпляров данной книги в библиотеке.

2.) Программа должна обеспечивать:

- хранение всех данных обо всех книгах в библиотеке в виде двусвязного списка, записи должны быть упорядочены по годам издания;
- добавление данных о книгах вновь поступивших в библиотеку;

- удаление данных о списываемой книге, название которой введено с клавиатуры;
- вывод информации обо всех книгах в библиотеке;
- по запросу выводится информация о книге, название которой введено с клавиатуры.

3.) Программа должна обеспечивать диалог с помощью меню.

Вариант 17

1.) Составить программу, которая содержит текущую информацию о книгах в библиотеке.

Сведения о книгах включают:

- фамилию и инициалы автора;
- название;
- год издания;
- количество экземпляров данной книги в библиотеке.

2.) Программа должна обеспечивать:

- хранение всех данных обо всех книгах в библиотеке в виде двусвязного списка, записи должны быть размещены в алфавитном порядке по фамилии автора;
- добавление данных о книгах, вновь поступивших в библиотеку;
- удаление данных о списываемой книге, название которой введено с клавиатуры;
- вывод информации обо всех книгах в библиотеке;
- по запросу выводится информация о книгах, изданных после года, введенного с клавиатуры.

3.) Программа должна обеспечивать диалог с помощью меню.

Вариант 18

1.) Составить программу, которая содержит текущую информацию о наличии товара на складе.

Сведения о товаре включают:

- название товара;
- название магазина, в котором продается товар;
- стоимость товара в рублях.
- номер партии товара.

2.) Программа должна обеспечивать:

- хранение всех данных обо всех товарах на складе в виде двусвязного списка, записи должны быть размещены в алфавитном порядке по названиям товаров;
- добавление данных о новых товарах;
- удаление данных о товаре, номер которого введен с клавиатуры;
- вывод сведений обо всех товарах;
- по запросу выводятся сведения о товаре, название которого введено с клавиатуры.

3.) Программа должна обеспечивать диалог с помощью меню.

Вариант 19

1.) Составить программу, которая содержит текущую информацию о наличии товара на складе.

Сведения о товаре включают:

- название товара;
- название магазина, в котором продается товар;
- стоимость товара в рублях.
- номер партии товара.

2.) Программа должна обеспечивать:

• хранение всех данных обо всех товарах на складе в виде двусвязного списка, записи должны быть размещены в алфавитном порядке по названиям магазинов;

• добавление данных о новых товарах;

• удаление данных о товаре, номер которого введен с клавиатуры;

• вывод сведений обо всех товарах;

• по запросу выводятся сведения о товарах, продающихся в магазине, название которого введено с клавиатуры.

3.) Программа должна обеспечивать диалог с помощью меню.

Вариант 20

1.) Составить программу, которая содержит текущую информацию о наличии товара на складе.

Сведения о товаре включают:

- название товара;
- название магазина, в котором продается товар;
- стоимость товара в рублях.
- номер партии товара.

2.) Программа должна обеспечивать:

• хранение всех данных обо всех товарах на складе в виде двусвязного списка, записи должны быть размещены в алфавитном порядке по названиям товаров;

• добавление данных о новых товарах;

• удаление данных о товаре, номер которого введен с клавиатуры;

• вывод сведений обо всех товарах;

• по запросу выводятся сведения о товарах, цена которых больше введенной с клавиатуры.

Программа должна обеспечивать диалог с помощью меню.

Вариант 21

1) Составить программу, которая содержит следующие сведения о коллекции картин в музее:

- название картины;
- фамилия автора;
- год написания;
- техника исполнения.

2) Программа должна обеспечивать:

- хранение всех данных обо всех картинах в музее в виде двусвязного списка, записи должны быть размещены в алфавитном порядке по фамилиям авторов;
- добавление данных о новых картинах;
- удаление данных о картине, автор и название которой введено с клавиатуры;
- вывод сведений обо всех картинах;
- по запросу выводятся сведения обо всех картинах, выполненных в определенной технике (вводится с клавиатуры).

3.) Программа должна обеспечивать диалог с помощью меню.

Вариант 22

1.) Составить программу, которая содержит следующие сведения

- название экскурсии, например, “Петергоф. Большой дворец”;
- дата экскурсии;
- время начала экскурсии.

2) Программа должна обеспечивать:

- хранение данных обо всех экскурсиях в виде двусвязного списка, записи должны быть упорядочены по дате и времени начала экскурсии;
- добавление данных о новых экскурсиях;
- удаление из списка информации об экскурсии, название которой введено с клавиатуры;
- вывод сведений обо всех экскурсиях;
- по запросу выводится информация об экскурсиях, из диапазона дат, введенных с клавиатуры.

3) Программа должна обеспечивать диалог с помощью меню.

Вариант 23

1) Составить программу, которая содержит следующие сведения о музеях Москвы:

- название музея, например, “Государственный Исторический музей”;
- тип музея, например, исторический, художественный (не менее 5-6 типов предусмотреть для выбора);
- адрес;
- краткое описание;
- сайт.

2) Программа должна обеспечивать:

- хранение данных обо всех музеях в виде двусвязного списка, записи должны быть упорядочены по названиям музеев;
- добавление данных о новых экскурсиях;
- удаление из списка информации об экскурсии, название которой введено с клавиатуры;
- вывод сведений обо всех экскурсиях;
- по запросу выводится информация о музеях, тип которых введен с клавиатуры.

3) Программа должна обеспечивать диалог с помощью меню.

Вариант 24

1) Составить программу, которая содержит следующие сведения о кораблях:

- название корабля, водоизмещение, порт приписки, капитан;
- название порта, страна;
- дата посещения порта, дата убытия, номер причала, цель посещения

2) Программа должна обеспечивать:

- хранение данных обо всех кораблях в виде двусвязного списка, записи должны быть упорядочены по названиям кораблей;
- добавление данных о кораблях;
- удаление из списка информации о корабле, название которого введено с клавиатуры;
- вывод сведений обо всех кораблях.

Один корабль может посещать несколько портов.

Определить, какие корабли посещали порт, название которого вводится с клавиатуры, с целью 'починки такелажа'.

Определить, с какой целью чаще всего заходят корабли в порт “Ялта”

Программа должна обеспечивать диалог с помощью меню.

Вариант 25

1) Составить программу, которая содержит следующие сведения о научных конференциях:

- ФИО ученого, организация, страна;
- название конференции, место проведения, дата;
- тип участия, тема доклада, публикация (да/нет).

Один ученый может участвовать в разных конференциях и с разными типами докладов (тип участия: доклад, сообщение, стендовый доклад, оргкомитет).

2) Программа должна обеспечивать:

- хранение данных обо всех ученых в виде двусвязного списка, записи должны быть упорядочены по ФИО;
- добавление данных об ученых;
- удаление из списка информации о конференции, название которой введено с клавиатуры;
- вывод сведений обо всех ученых;
- по запросу выводится список ученых, имевших публикации в году, введенном с клавиатуры, с указанием числа публикаций для каждого.

3) Программа должна обеспечивать диалог с помощью меню.