

## Лабораторная работа №7

### Линейный динамический список

**Пример 1.** Написать функцию, создающую односвязный список для хранения целочисленных данных, состоящий из одного элемента со значением, вводимым с клавиатуры.

```
#include<stdio.h>
struct Element{    // структура, задающая элемент списка
    int data;        // поле для хранения данных
    Element *next;    // указатель на следующий элемент
} *start, *p;        //определяем глобальные указатели на начало списка и текущий элемент
void create_list(int a){
    p = new Element;
    p->data = a;
    p->next = NULL;
    start = p;
    return;
}
int main(){
    int a;
    printf("Введите значение для первого элемента, создаваемого списка: ");
    scanf("%i",&a);
    create_list(a);
    printf("Список создан, единственный элемент: %i \n",start->data);
    return 0;
}
```

**Пример 2.** Написать функцию, добавляющую элемент в уже существующий односвязный список следом за текущим элементом. Значение нового элемента передается как параметр функции.

```
void new_element(int a){
    Element *q;
    q = new Element;
    q->data = a;
    q->next = p->next;
    p->next = q;
    return;
}
```

**Пример 3.** Написать функцию, удаляющую текущий элемент односвязного списка.

```
void remove_element(){
    if(p == start){
        start = p->next;
        delete p;
        p = start; //текущий - это теперь следующий за удаленным
        return;
    }
    Element *q;
    q = start;
    while(q->next != p){
        q = q->next;
    }
    q->next = p->next;
    delete p;
    p = q; //текущий - это теперь предшествующий удаленному
    return;
}
```

**Пример 4.** Пример циклического списка (*задача Иосифа*)

Предположим,  $N$  человек решили выбрать главаря. Для этого они встали в круг и стали удалять каждого  $M$ -го человека в определенном направлении отсчета, смыкая ряды после каждого удаления. Задача состоит в определении, кто останется последним (потенциальный лидер с математическими способностями заранее определит выигрышную позицию в круге).

Для представления людей, расставленных в круг, построим односвязный *циклический* список, где каждый элемент (человек) содержит ссылку на соседний элемент против хода часовой стрелки. Сначала создается список элементов от 1 до  $N$ . Для этого создается циклический список с единственным узлом для участника 1, затем вставляются узлы для участников от 2 до  $N$  с помощью цикла. Затем в списке отсчитывается  $(M - 1)$  элемент и удаляется следующий. Этот процесс продолжается до тех пор, пока не останется только один узел (который будет указывать на самого себя).

```

struct Node{
    int item;
    Node *next;
};

int main(){
    int i, N, M;
    printf ("Сколько в банде\n");
    scanf ("%i",&N);
    printf ("Какой по счёту\n");
    scanf ("%i",&M);

    Node *t = new Node;
    t->item =1;
    t->next = t;

    Node *x = t;
    for (i = 2; i <= N; i++){
        Node *q = new Node;
        q->item = i;
        q->next = t;
        x->next = q;
        x = q;
    }

    while (x != x->next){
        for (i = 1; i < M; i++)
            x = x->next;
        x->next = x->next->next;
    }

    printf ("Остался %i - ый человек \n", x->item);
    return 0; // найдите в функции main() ОШИБКУ!!!
}

```

### Задание 7.

1. Напишите программу, создающую линейный динамический список, указанный в задании. Значения элементов вводятся с консоли. Вам понадобятся функции «создать», «добавить», «удалить список», «просмотр списка», а также переменная-указатель на начало списка.

2. Реализуйте указанную в задании функцию и продемонстрируйте ее работу.

- 7.1. Односвязный список. Поиск первого элемента в списке, совпадающего с заданным числом.
- 7.2. Односвязный список. Поиск минимального элемента в списке.
- 7.3. Односвязный список. Поиск максимального элемента в списке.
- 7.4. Односвязный циклический список. Поиск первого элемента в списке, совпадающего с заданным числом.
- 7.5. Односвязный циклический список. Поиск минимального элемента в списке.
- 7.6. Односвязный циклический список. Поиск максимального элемента в списке.
- 7.7. Односвязный список. Значение текущего элемента в списке (пользователь указывает номер элемента).
- 7.8. Односвязный список. Значение элемента, предшествующего текущему (пользователь указывает номер элемента).
- 7.9. Односвязный список. Сумма значений  $n$  элементов, начиная с текущего (пользователь указывает номер элемента).
- 7.10. Двусвязный список. Просмотр списка справа налево.
- 7.11. Двусвязный список. Удаление  $n$  элементов, начиная с текущего элемента (пользователь указывает номер элемента).
- 7.12. Двусвязный список. Сумма значений  $n$  элементов, начиная с текущего (пользователь указывает номер элемента).
- 7.13. Двусвязный список. Удаление  $n$  элементов, начиная с текущего элемента (пользователь указывает номер элемента).
- 7.14. Односвязный список. Сравнения элементов двух списков с занесением одинаковых в третий список.
- 7.15. Односвязный циклический список. Сравнения элементов двух списков с занесением одинаковых в третий список.