

**Московский Авиационный Институт
(Национальный Исследовательский Университет)**

**Факультет информационных технологий и прикладной
математики**

**Кафедра вычислительной математики и
программирования**

**Курсовой проект VIII
по курсу «Практикум на ЭВМ»
II семестр**

«Линейные списки»

Студент	Сыроежкин Кирилл Геннадьевич
Группа	М8О-104Б-18
Руководитель	Доцент кафедры 806 Никулин С.П
Оценка	
Дата	

Москва 2019.

1. Постановка задачи.

Составить и отладить программу на языке Си для обработки линейного списка заданной организации с отображением списка на динамические структуры. Навигацию по списку следует реализовать с применением итераторов. Предусмотреть выполнение одного нестандартного и четырех стандартных действий:

- Печать списка.
- Вставка нового элемента в список.
- Удаление элемента из списка.
- Подсчет длины списка.

Вариант

$N=18$

Тип элемента списка: (определяется как $\text{номер_группы} \% 8 + 1$)

целый

Вид списка: (определяется как $(N \div 2) \% 6 + 1$)

Кольцевой двунаправленный (без использования указателей)

Нестандартное действие: (определяется как $N \% 15 + 1$)

Обменять $k-1$ и $k+1$ элемент

2. Общие сведения о списках

Кольцевой список — это структура данных, представляющая собой последовательность элементов, последний элемент которой содержит указатель на первый элемент списка, а первый (в случае двунаправленного списка) — на последний. Основная особенность такой организации состоит в том, что в этом списке нет элементов, содержащих пустые указатели, и, следовательно, нельзя выделить крайние элементы.

Каждый узел двунаправленного (двусвязного) циклического списка (ДЦС) содержит два поля указателей — на следующий и на предыдущий узлы. Указатель на предыдущий узел корня списка содержит адрес последнего узла. Указатель на следующий узел последнего узла содержит адрес корня списка.

3. Общие сведения о программе

Имена файлов:

c8.c - программа;

Программное и аппаратное обеспечения для запуска данной программы на ПК не ограничено в выборе. Операционная система семейства Unix - Ubuntu. Язык программирования Си. Строк в программе prog.c: 97, dump.c: 28. В заголовочном файле pass.h:15.

4. Описание логической структуры программы

Так как заданию требуется реализовать список без использования указателей, то в структуре храним не ссылки на элементы, а индексы (следующего/предыдущего)

Основные функции:

Для инициализации списка используется функция **list_create** (структура список), которая обнуляет все элементы и индексы внутри структуры.

list_update (список, размер) устанавливает связи в списке или обновляет их, если был добавлен новый элемент.

list_insert_elem (список, позиция, элемент) вставляет элемент в список на указанную позицию

list_delete_elem(список, позиция) удаляет элемент на указанной позиции

list_print (список) печатает список

list_destroy (список) обнуляет все элементы списка.

list_exchange (список, k) обменивает k+1 и k-1 элемент

Итераторы:

list_next_elem - индекс следующего элемента

list_prev_elem - индекс предыдущего элемента

list_fetch - дает значение элемента по индексу

list_store - присваивает значение элемента по индексу

list_first_elem - индекс первого элемента

list_last_elem - индекс последнего элемента

list_size - длина списка

list_empty - проверка на пустоту списка

5. Входные данные

Программа должна получать с клавиатуры максимальный размер списка, а затем целые числа в зависимости от пунктов меню

6. Выходные данные

Результат работы программы может меняться в зависимости от выбранного пунктов меню:

- 1) Вставить элемент в конец списка;
- 2) Вставка элемента перед определенным элементом;
- 3) Удаление элемента из списка;
- 4) Обменять местами $k+1$ и $k-1$ элемент;
- 5) Печать списка;
- 6) Длина списка;
- 7) Очистить список.

7. Программа на СИ

```
#include <stdio.h>
#include <stdbool.h>
int max_size;
typedef struct
{
    int num;
    int next_index;
    int prev_index;
    int last_index;
    int first_index;
    int size;
} list;
int list_create(list* l) //Объявление списка
{
    for (int i = 0; i < max_size; i++)
        l[i].num=0;
```

```

    l->size=0;
    l->last_index=l->first_index=0;
}
int list_update(list *l, int size) //обновление длины списка
{
    int c=0;
    for (int i = 0; i < size; i++)
    {
        c++;
        if (size==1)
        {
            l[i].next_index=l[i].prev_index=i;
            l->first_index=i;
            l->last_index=i;
        }
        else if (i==0)
        {
            l[i].next_index=i+1;
            l[i].prev_index=size-1;
            l->first_index=i;
        }
        else if (i==size-1)
        {
            l[i].next_index=l->first_index;
            l[i].prev_index=i-1;
            l->last_index=i;
        }
        else
        {
            l[i].next_index=i+1;
            l[i].prev_index=i-1;

```

```

        }
    }
    l->size=c;
}

int list_next_elem(list* l, int i) //Итератор следующего
элемента
{
    i=l[i].next_index;
    return i;
}

int list_prev_elem(list* l, int i) // Итератор предыдущего
элемента
{
    i=l[i].prev_index;
    return i;
}

int list_fetch(list* l, int i) // Итератор, дающий значение
по индексу
{
    return l[i].num;
}

void list_store(list* l, int i, int t) // Итератор,
присваивающий значение по индексу
{
    l[i].num=t;
}

int list_first_elem(list* l) // Итератор, возвращающий
первый элемент
{
    return l->first_index;
}

int list_last_elem(list* l) // Итератор, возвращающий
последний элемент

```

```

{
    return l->last_index;
}
int list_size(list* l) //Итератор, возвращающий длину списка
{
    return l->size;
}
bool list_empty(list* l) //Итератор, проверяющий на пустоту
список
{
    if (l->size==0)
        return true;
    return false;
}
bool list_insert_elem(list *l, int i, int t) //вставка
элемента в любую часть списка
{
    if (list_size(l)==max_size)
    {
        printf("\nПривышен      максимальный      размер
списка!\n");
        return false;
    }
    if (i>list_size(l))
    {
        printf("\nЭлемента    с    таким    индексом    не
существует!\n");
        return false;
    }
    list_update(l, list_size(l)+1);
    if (i==0)
        list_store(l,list_size(l)-1, t);
    else

```

```

    {
        int c=i-1;
        for (int p=list_size(l)-1; p>=i; p--)
            list_store(l, p, list_fetch(l,
            list_prev_elem(l,p)));
        list_store(l,c,t);
    }
    return true;
}

bool list_delete_elem(list* l, int i) //Удаление элемента
из списка
{
    if (i>list_size(l))
    {
        printf("\nЭлемента с таким индексом не
        существует!\n");
        return false;
    }
    for (i; i<list_size(l); i++)
        list_store(l,i,list_fetch(l,list_next_elem(l,i)))
    ;
    list_store(l, list_size(l)-1, 0);
    list_update(l, list_size(l)-1);
    return true;
}

void list_print(list* l) //Печать списка
{
    printf("\n");
    for (int i = 0; i < list_size(l); i++)
    {
        printf(" %d ", list_fetch(l,i));
    }
    printf("\n");
}

```



```

}
void list_exchange(list *l, int k) //обмен двумя элементами
{
    int t;
    t=list_fetch(l, list_prev_elem(l,k));
    list_store(l, list_prev_elem(l,k), list_fetch(l,
list_next_elem(l,k)));
    list_store(l, list_next_elem(l,k), t);
}
void list_destroy(list *l) //очистить список
{
    for (int i=0; i<max_size; i++)
        list_store(l, i, 0);
    l->size=0;
}
void print_menu()
{
    printf("\n1.Вставить элемент в конец списка \n2.Вставка
элемента перед определенным элементом.\n3.Удаление
элемента из списка.\n4.Обменять местами k+1 и k-1
элемент.\n5.Печать списка.\n6.Длина
списка.\n7.Очистить список.\n\nВыберите действие: ");
}
int main()
{
    printf("Введите максимальный размер кольцевого
двусвязного списка: ");
    scanf("%d", &max_size);
    list l[max_size];
    int act, elem, position;
    list_create(l);
    print_menu();
    while(scanf("%d", &act)!=EOF)

```

```

{
    switch(act)
    {
        case 1:
            printf("\nЭлемент: ");
            scanf("%d", &elem);
            list_insert_elem(l, 0, elem);
            break;
        case 2:
            printf("\nВыберите      перед      каким
элементов вставить нужный(от 1): ");
            scanf("%d", &position);
            printf("\nЭлемент: ");
            scanf("%d", &elem);
            list_insert_elem(l, position, elem);
            break;
        case 3:
            printf("\nВыберите на какой позиции
удалить элемент(от 1): ");
            scanf("%d", &position);
            list_delete_elem(l, position-1);
            break;
        case 4:
            printf("\nk(от 1):");
            scanf("%d", &position);
            list_exchange(l, position-1);
            break;
        case 5:
            list_print(l);
            break;
        case 6:
            printf("\n%d\n", list_size(l));
    }
}

```

```

        break;
    case 7:
        list_destroy(1);
        break;
    }
    print_menu();
}
printf("\n");
}

```

8. Демонстрация работы программы

kircatle@DESKTOP70J5NO3:/mnt/c/Users/joker/Desktop/LabsMAI/SecondSem/course8\$ gcc c8.c

kircatle@DESKTOP-

70J5NO3:/mnt/c/Users/joker/Desktop/LabsMAI/SecondSem/course8\$./a.out

ТЕСТ 1

Введите максимальный размер кольцевого двусвязного списка: 8

1. Вставить элемент в конец списка
2. Вставка элемента перед определенным элементом.
3. Удаление элемента из списка.
4. Обменять местами $k+1$ и $k-1$ элемент.
5. Печать списка.
6. Длина списка.
7. Очистить список.

Выберите действие: 1

Элемент: 12

1. Вставить элемент в конец списка
2. Вставка элемента перед определенным элементом.
3. Удаление элемента из списка.
4. Обменять местами $k+1$ и $k-1$ элемент.
5. Печать списка.
6. Длина списка.
7. Очистить список.

Выберите действие: 2

Выберите перед каким элементов вставить нужный(от 1):
1

Элемент: 5

1. Вставить элемент в конец списка
2. Вставка элемента перед определенным элементом.
3. Удаление элемента из списка.
4. Обменять местами $k+1$ и $k-1$ элемент.
5. Печать списка.

6.Длина списка.

7.Очистить список.

Выберите действие: 1

Элемент: 52

1.Вставить элемент в конец списка

2.Вставка элемента перед определенным элементом.

3.Удаление элемента из списка.

4.Обменять местами $k+1$ и $k-1$ элемент.

5.Печать списка.

6.Длина списка.

7.Очистить список.

Выберите действие: 5

5 12 52

1.Вставить элемент в конец списка

2.Вставка элемента перед определенным элементом.

3.Удаление элемента из списка.

4.Обменять местами $k+1$ и $k-1$ элемент.

5.Печать списка.

6.Длина списка.

7.Очистить список.

Выберите действие: 1

Элемент: 24

1.Вставить элемент в конец списка

2.Вставка элемента перед определенным элементом.

3.Удаление элемента из списка.

4.Обменять местами $k+1$ и $k-1$ элемент.

5.Печать списка.

6.Длина списка.

7.Очистить список.

Выберите действие: 2

Выберите преред каким элементов вставить нужный(от 1):
3

Элемент: 44

1.Вставить элемент в конец списка

2.Вставка элемента перед определенным элементом.

3.Удаление элемента из списка.

4.Обменять местами $k+1$ и $k-1$ элемент.

5.Печать списка.

6.Длина списка.

7.Очистить список.

Выберите действие: 2

Выберите преред каким элементов вставить нужный(от 1):
6

Элемент: 3

Элемента с таким индексом не существует!

1.Вставить элемент в конец списка

2.Вставка элемента перед определенным элементом.

3.Удаление элемента из списка.

4.Обменять местами $k+1$ и $k-1$ элемент.

5.Печать списка.

6.Длина списка.

7.Очистить список.

Выберите действие: 5

5 12 44 52 24

1. Вставить элемент в конец списка
2. Вставка элемента перед определенным элементом.
3. Удаление элемента из списка.
4. Обменять местами $k+1$ и $k-1$ элемент.
5. Печать списка.
6. Длина списка.
7. Очистить список.

Выберите действие: 6

5

1. Вставить элемент в конец списка
2. Вставка элемента перед определенным элементом.
3. Удаление элемента из списка.
4. Обменять местами $k+1$ и $k-1$ элемент.
5. Печать списка.
6. Длина списка.
7. Очистить список.

Выберите действие: 2

Выберите перед каким элементов вставить нужный(от 1):

3

Элемент: 5

1. Вставить элемент в конец списка
2. Вставка элемента перед определенным элементом.
3. Удаление элемента из списка.
4. Обменять местами $k+1$ и $k-1$ элемент.
5. Печать списка.
6. Длина списка.
7. Очистить список.

Выберите действие: 2

Выберите перед каким элементов вставить нужный(от 1):
5

Элемент: 11

1. Вставить элемент в конец списка
2. Вставка элемента перед определенным элементом.
3. Удаление элемента из списка.
4. Обменять местами $k+1$ и $k-1$ элемент.
5. Печать списка.
6. Длина списка.
7. Очистить список.

Выберите действие: 5

5 12 5 44 11 52 24

1. Вставить элемент в конец списка
2. Вставка элемента перед определенным элементом.
3. Удаление элемента из списка.
4. Обменять местами $k+1$ и $k-1$ элемент.
5. Печать списка.
6. Длина списка.
7. Очистить список.

Выберите действие: 2

Выберите перед каким элементом вставить нужный(от 1):
7

Элемент: 11

1. Вставить элемент в конец списка
2. Вставка элемента перед определенным элементом.
3. Удаление элемента из списка.
4. Обменять местами $k+1$ и $k-1$ элемент.
5. Печать списка.

6.Длина списка.

7.Очистить список.

Выберите действие: 5

5 12 5 44 11 52 11 24

1.Вставить элемент в конец списка

2.Вставка элемента перед определенным элементом.

3.Удаление элемента из списка.

4.Обменять местами $k+1$ и $k-1$ элемент.

5.Печать списка.

6.Длина списка.

7.Очистить список.

Выберите действие: 1

Элемент: 90

Привышен максимальный размер списка!

1.Вставить элемент в конец списка

2.Вставка элемента перед определенным элементом.

3.Удаление элемента из списка.

4.Обменять местами $k+1$ и $k-1$ элемент.

5. Печать списка.
6. Длина списка.
7. Очистить список.

Выберите действие: 4

k(от 1):3

1. Вставить элемент в конец списка
2. Вставка элемента перед определенным элементом.
3. Удаление элемента из списка.
4. Обменять местами $k+1$ и $k-1$ элемент.
5. Печать списка.
6. Длина списка.
7. Очистить список.

Выберите действие: 5

5 44 5 12 11 52 11 24

1. Вставить элемент в конец списка
2. Вставка элемента перед определенным элементом.
3. Удаление элемента из списка.
4. Обменять местами $k+1$ и $k-1$ элемент.

5. Печать списка.
6. Длина списка.
7. Очистить список.

Выберите действие: 4

$k(\text{от } 1):1$

1. Вставить элемент в конец списка
2. Вставка элемента перед определенным элементом.
3. Удаление элемента из списка.
4. Обменять местами $k+1$ и $k-1$ элемент.
5. Печать списка.
6. Длина списка.
7. Очистить список.

Выберите действие: 5

5 24 5 12 11 52 11 44

1. Вставить элемент в конец списка
2. Вставка элемента перед определенным элементом.
3. Удаление элемента из списка.
4. Обменять местами $k+1$ и $k-1$ элемент.

5.Печать списка.

6.Длина списка.

7.Очистить список.

Выберите действие: 3

Выберите на какой позиции удалить элемент(от 1): 8

1.Вставить элемент в конец списка

2.Вставка элемента перед определенным элементом.

3.Удаление элемента из списка.

4.Обменять местами $k+1$ и $k-1$ элемент.

5.Печать списка.

6.Длина списка.

7.Очистить список.

Выберите действие: 3

Выберите на какой позиции удалить элемент(от 1): 7

1.Вставить элемент в конец списка

2.Вставка элемента перед определенным элементом.

3.Удаление элемента из списка.

4.Обменять местами $k+1$ и $k-1$ элемент.

5.Печать списка.

6.Длина списка.

7.Очистить список.

Выберите действие: 5

5 24 5 12 11 52

1.Вставить элемент в конец списка

2.Вставка элемента перед определенным элементом.

3.Удаление элемента из списка.

4.Обменять местами $k+1$ и $k-1$ элемент.

5.Печать списка.

6.Длина списка.

7.Очистить список.

Выберите действие: 3

Выберите на какой позиции удалить элемент(от 1): 3

1.Вставить элемент в конец списка

2.Вставка элемента перед определенным элементом.

3.Удаление элемента из списка.

4.Обменять местами $k+1$ и $k-1$ элемент.

5. Печать списка.
6. Длина списка.
7. Очистить список.

Выберите действие: 5

5 24 12 11 52

1. Вставить элемент в конец списка
2. Вставка элемента перед определенным элементом.
3. Удаление элемента из списка.
4. Обменять местами $k+1$ и $k-1$ элемент.
5. Печать списка.
6. Длина списка.
7. Очистить список.

Выберите действие: 7

1. Вставить элемент в конец списка
2. Вставка элемента перед определенным элементом.
3. Удаление элемента из списка.
4. Обменять местами $k+1$ и $k-1$ элемент.
5. Печать списка.
6. Длина списка.

7.Очистить список.

Выберите действие: 5

- 1.Вставить элемент в конец списка
- 2.Вставка элемента перед определенным элементом.
- 3.Удаление элемента из списка.
- 4.Обменять местами $k+1$ и $k-1$ элемент.
- 5.Печать списка.
- 6.Длина списка.
- 7.Очистить список.

Выберите действие:

ТЕСТ 2

Введите максимальный размер кольцевого двусвязного списка: 4

- 1.Вставить элемент в конец списка
- 2.Вставка элемента перед определенным элементом.
- 3.Удаление элемента из списка.
- 4.Обменять местами $k+1$ и $k-1$ элемент.
- 5.Печать списка.
- 6.Длина списка.

7.Очистить список.

Выберите действие: 1

Элемент: 3

- 1.Вставить элемент в конец списка
- 2.Вставка элемента перед определенным элементом.
- 3.Удаление элемента из списка.
- 4.Обменять местами $k+1$ и $k-1$ элемент.
- 5.Печать списка.
- 6.Длина списка.
- 7.Очистить список.

Выберите действие: 1

Элемент: 5

- 1.Вставить элемент в конец списка
- 2.Вставка элемента перед определенным элементом.
- 3.Удаление элемента из списка.
- 4.Обменять местами $k+1$ и $k-1$ элемент.
- 5.Печать списка.
- 6.Длина списка.

7.Очистить список.

Выберите действие: 1

Элемент: 8

- 1.Вставить элемент в конец списка
- 2.Вставка элемента перед определенным элементом.
- 3.Удаление элемента из списка.
- 4.Обменять местами $k+1$ и $k-1$ элемент.
- 5.Печать списка.
- 6.Длина списка.
- 7.Очистить список.

Выберите действие: 2

Выберите преред каким элементов вставить нужный(от 1):
3

Элемент: 11

- 1.Вставить элемент в конец списка
- 2.Вставка элемента перед определенным элементом.
- 3.Удаление элемента из списка.
- 4.Обменять местами $k+1$ и $k-1$ элемент.

5.Печать списка.

6.Длина списка.

7.Очистить список.

Выберите действие: 5

3 5 11 8

1.Вставить элемент в конец списка

2.Вставка элемента перед определенным элементом.

3.Удаление элемента из списка.

4.Обменять местами $k+1$ и $k-1$ элемент.

5.Печать списка.

6.Длина списка.

7.Очистить список.

Выберите действие: 2

Выберите преред каким элементов вставить нужный(от 1):
3

Элемент: 44

Привышен максимальный размер списка!

1. Вставить элемент в конец списка
2. Вставка элемента перед определенным элементом.
3. Удаление элемента из списка.
4. Обменять местами $k+1$ и $k-1$ элемент.
5. Печать списка.
6. Длина списка.
7. Очистить список.

Выберите действие: 4

$k(\text{от } 1):3$

1. Вставить элемент в конец списка
2. Вставка элемента перед определенным элементом.
3. Удаление элемента из списка.
4. Обменять местами $k+1$ и $k-1$ элемент.
5. Печать списка.
6. Длина списка.
7. Очистить список.

Выберите действие: 5

3 8 11 5

1. Вставить элемент в конец списка
2. Вставка элемента перед определенным элементом.
3. Удаление элемента из списка.
4. Обменять местами $k+1$ и $k-1$ элемент.
5. Печать списка.
6. Длина списка.
7. Очистить список.

Выберите действие: 3

Выберите на какой позиции удалить элемент(от 1): 1

1. Вставить элемент в конец списка
2. Вставка элемента перед определенным элементом.
3. Удаление элемента из списка.
4. Обменять местами $k+1$ и $k-1$ элемент.
5. Печать списка.
6. Длина списка.
7. Очистить список.

Выберите действие: 5

8 11 5

1. Вставить элемент в конец списка
2. Вставка элемента перед определенным элементом.
3. Удаление элемента из списка.
4. Обменять местами $k+1$ и $k-1$ элемент.
5. Печать списка.
6. Длина списка.
7. Очистить список.

Выберите действие: 7

1. Вставить элемент в конец списка
2. Вставка элемента перед определенным элементом.
3. Удаление элемента из списка.
4. Обменять местами $k+1$ и $k-1$ элемент.
5. Печать списка.
6. Длина списка.
7. Очистить список.

Выберите действие:

Литература

Методические указания к выполнению курсовых работ. Зайцев В. Е.

https://learnc.info/c/text_files.html

<http://www.sbp-program.ru/c/sbp-file-c.htm>

[http://files.mai.ru/site/priem/documents/orders/2017/111_03.08.2017.
pdf](http://files.mai.ru/site/priem/documents/orders/2017/111_03.08.2017.pdf)