**Отчет по лабораторной работе № 8** по курсу

«Языки и методы программирования»

Студент группы М8О-112Б-21 Орешкин Максим Алексеевич, № по списку \_\_\_13\_\_

Контакты www, e-mail: maks-oreh03@mail.ru

Работа выполнена: « » \_\_\_\_\_\_ 202\_ г.

Преподаватель: доцент каф. 806 \_Никулин С.П\_\_\_\_\_\_\_\_\_\_\_\_\_

Входной контроль знаний с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Отчет сдан « » \_\_\_\_\_\_\_\_\_202\_ г., итоговая оценка \_\_\_\_\_

Подпись преподавателя \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ь преподавателя \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Тема:** Линейные списки
2. **Цель работы:** Составить и отладить программу на языке Си для обработки линейного списка заданной организации с отображением списка на динамическую структуру.
3. **Задание** : Тип элемента списка 6 – Комплексный, Вид списка 1 – кольцевой однонаправленный, нестандартное действие 14 – Выполнить попарный обмен значениями элементов списка.
4. **Оборудование** :

ЭВМ , процессор , имя узла сети с ОП ГБ,

НМД ГБ. Терминал адрес . Принтер

Другие устройства

*Оборудование ПЭВМ студента, если использовалось:*

 Процессор intel core i7, с   ОП 8 ГБ, НМД  120832 МБ. Монитор  15,6 /1920\*1820

Другие устройства

1. **Программное обеспечение:**

Операционная система семейства , наименование версия интерпретатор команд версия

Система программирования \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ версия \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Редактор текстов версия

Утилиты операционной системы

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Прикладные системы и программы

Местонахождение и имена файлов программ и данных

*Программное обеспечение ЭВМ студента, если использовалось:*

Операционная система семейства Unix , наименование Ubuntu версия 20.04

интерпретатор команд bash версия 4.4.18

Система программирования версия

Редактор текстов vim версия 8.1

Утилиты операционной системы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Прикладные системы и программы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Местонахождение и имена файлов программ и данных на домашнем компьютере \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Идея, метод, алгоритм**  решения задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальные спецификации с пред- и постусловиями)

Просмотреть перечень заданий и реализовать функции, которые просят.

1. **Сценарий выполнения работы** [план работы, первоначальный текст программы в черновике (можно на отдельном листе) и тесты либо соображения по тестированию].

#include <stdlib.h>

#include <stdio.h>

typedef struct complex {

double a,b;

} complex;

complex mul(double x, double y)

{

complex t;

t.a = x;

t.b = y;

return t;

}

typedef struct Node{

complex data;

struct Node \* next;

} Node;

Node \*init(complex a)

{

Node \*lst;

lst = (Node\*) malloc(sizeof(Node));

lst->data = a;

lst->next = lst;

return lst;

}

Node \* addelem(Node \*lst, complex number)

{

if(lst==NULL) {printf("Узел еще не создан"); return NULL;}

else{

Node \* temp,\*p;

temp = (Node \*) malloc(sizeof(lst));

p = lst->next;

lst->next=temp;

temp->data = number;

temp->next = p;

return temp;

}

}

Node \* deletelem(Node \*lst, complex t)

{

Node \*temp;

temp = lst;

while(temp->next != lst)

{

temp = temp->next;

}

temp->next = lst->next;

free(lst);

return temp;

}

void print(Node \*lst){

Node \*p = lst;

do{

printf("%f + %fi\n", p->data.a,p->data.b);

p=p->next;

}while(p!=lst);

}

int count(Node \*lst){

Node \*p = lst;

int count=0;

do{

count++;

p=p->next;

}while(p!=lst);

return count;

}

Node \*obmen(Node\*lst, complex A, complex B ){

Node \*p = lst;

Node \*tmp1,\*tmp2;

do

{

if(p->data.a==A.a && p->data.b==A.b) {p->data.a = B.a;p->data.b = B.b;}

else if(p->data.a==B.a && p->data.b==B.b) {p->data.a = A.a;p->data.b = A.b;}

p=p->next;

}while(p!=lst);

return p;

}

int main(void)

{

Node \*lst = NULL;

int input;

while(1)

{

printf("1 Создание списка\n2 Вставка нового элемента в список\n3 Печать списка\n4 Удаление элемента из списка\n5 подсчет длины списка\n6 Попарный обмен значениями\n7 Выход\n");

scanf("%d", &input);

switch(input)

{

case 1:

printf("Запись числа в узел...\n");

printf("Введите целую и мнимую часть числа\n");

double a,b;

scanf("%lf %lf", &a,&b);

lst = init(mul(a,b));

break;

case 2:

printf("Запись числа в узел...\n");

printf("Введите целую и мнимую часть числа\n");

scanf("%lf %lf",&a,&b);

lst = addelem(lst,mul(a,b));

break;

case 3:

print(lst);

break;

case 4:

printf("Введите целую и мнимую часть числа для удаления элемента из узла\n");

scanf("%lf %lf", &a,&b);

lst = deletelem(lst,mul(a,b));

break;

case 5:

printf("Количество элементов = %d\n", count(lst));

break;

case 6:

printf("Введите целую и мнимую часть для двух элементов, которые хотите поменять местами\n");

double c,d;

scanf("%lf %lf %lf %lf", &a, &b, &c, &d);

lst = obmen(lst, mul(a,b), mul(c,d));

break;

case 7:

exit(0);

break;

default:

printf("Incorrect unput\n");

break;

}

}

}

*Пункты 1-7 отчета составляются строго до начала лабораторной работы.*

*Допущен к выполнению работы.*  **Подпись преподавателя \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. **Распечатка протокола**  (подклеить листинг окончательного варианта программы с тестовыми примерами, подписанный преподавателем).

maxim@vb:~$ gcc VIII.c

maxim@vb:~$ ./a.out

1 Создание списка

2 Вставка нового элемента в список

3 Печать списка

4 Удаление элемента из списка

5 подсчет длины списка

6 Попарный обмен значениями

7 Выход

1

Запись числа в узел...

Введите целую и мнимую часть числа

1

2.234

1 Создание списка

2 Вставка нового элемента в список

3 Печать списка

4 Удаление элемента из списка

5 подсчет длины списка

6 Попарный обмен значениями

7 Выход

2

Запись числа в узел...

Введите целую и мнимую часть числа

2

3

1 Создание списка

2 Вставка нового элемента в список

3 Печать списка

4 Удаление элемента из списка

5 подсчет длины списка

6 Попарный обмен значениями

7 Выход

2

Запись числа в узел...

Введите целую и мнимую часть числа

4

5.32

1 Создание списка

2 Вставка нового элемента в список

3 Печать списка

4 Удаление элемента из списка

5 подсчет длины списка

6 Попарный обмен значениями

7 Выход

3

4.000000 + 5.320000i

1.000000 + 2.234000i

2.000000 + 3.000000i

1 Создание списка

2 Вставка нового элемента в список

3 Печать списка

4 Удаление элемента из списка

5 подсчет длины списка

6 Попарный обмен значениями

7 Выход

4

Введите целую и мнимую часть числа для удаления элемента из узла

2

3

1 Создание списка

2 Вставка нового элемента в список

3 Печать списка

4 Удаление элемента из списка

5 подсчет длины списка

6 Попарный обмен значениями

7 Выход

3

2.000000 + 3.000000i

1.000000 + 2.234000i

1 Создание списка

2 Вставка нового элемента в список

3 Печать списка

4 Удаление элемента из списка

5 подсчет длины списка

6 Попарный обмен значениями

7 Выход

5

Количество элементов = 2

1 Создание списка

2 Вставка нового элемента в список

3 Печать списка

4 Удаление элемента из списка

5 подсчет длины списка

6 Попарный обмен значениями

7 Выход

2

Запись числа в узел...

Введите целую и мнимую часть числа

4

5

1 Создание списка

2 Вставка нового элемента в список

3 Печать списка

4 Удаление элемента из списка

5 подсчет длины списка

6 Попарный обмен значениями

7 Выход

5

Количество элементов = 3

1 Создание списка

2 Вставка нового элемента в список

3 Печать списка

4 Удаление элемента из списка

5 подсчет длины списка

6 Попарный обмен значениями

7 Выход

3

4.000000 + 5.000000i

1.000000 + 2.234000i

2.000000 + 3.000000i

1 Создание списка

2 Вставка нового элемента в список

3 Печать списка

4 Удаление элемента из списка

5 подсчет длины списка

6 Попарный обмен значениями

7 Выход

2

Запись числа в узел...

Введите целую и мнимую часть числа

6

7

1 Создание списка

2 Вставка нового элемента в список

3 Печать списка

4 Удаление элемента из списка

5 подсчет длины списка

6 Попарный обмен значениями

7 Выход

2

Запись числа в узел...

Введите целую и мнимую часть числа

5.456

6.789

1 Создание списка

2 Вставка нового элемента в список

3 Печать списка

4 Удаление элемента из списка

5 подсчет длины списка

6 Попарный обмен значениями

7 Выход

3

5.456000 + 6.789000i

1.000000 + 2.234000i

2.000000 + 3.000000i

4.000000 + 5.000000i

6.000000 + 7.000000i

1 Создание списка

2 Вставка нового элемента в список

3 Печать списка

4 Удаление элемента из списка

5 подсчет длины списка

6 Попарный обмен значениями

7 Выход

5

Количество элементов = 5

1 Создание списка

2 Вставка нового элемента в список

3 Печать списка

4 Удаление элемента из списка

5 подсчет длины списка

6 Попарный обмен значениями

7 Выход

6

Введите целую и мнимую часть для двух элементов, которые хотите поменять местами

2

3

6

7

1 Создание списка

2 Вставка нового элемента в список

3 Печать списка

4 Удаление элемента из списка

5 подсчет длины списка

6 Попарный обмен значениями

7 Выход

3

5.456000 + 6.789000i

1.000000 + 2.234000i

6.000000 + 7.000000i

4.000000 + 5.000000i

2.000000 + 3.000000i

1 Создание списка

2 Вставка нового элемента в список

3 Печать списка

4 Удаление элемента из списка

5 подсчет длины списка

6 Попарный обмен значениями

7 Выход

6

Введите целую и мнимую часть для двух элементов, которые хотите поменять местами

4

5

5.456

6.789

1 Создание списка

2 Вставка нового элемента в список

3 Печать списка

4 Удаление элемента из списка

5 подсчет длины списка

6 Попарный обмен значениями

7 Выход

3

4.000000 + 5.000000i

1.000000 + 2.234000i

6.000000 + 7.000000i

5.456000 + 6.789000i

2.000000 + 3.000000i

1 Создание списка

2 Вставка нового элемента в список

3 Печать списка

4 Удаление элемента из списка

5 подсчет длины списка

6 Попарный обмен значениями

7 Выход

5

Количество элементов = 5

1 Создание списка

2 Вставка нового элемента в список

3 Печать списка

4 Удаление элемента из списка

5 подсчет длины списка

6 Попарный обмен значениями

7 Выход

4

Введите целую и мнимую часть числа для удаления элемента из узла

4

5

1 Создание списка

2 Вставка нового элемента в список

3 Печать списка

4 Удаление элемента из списка

5 подсчет длины списка

6 Попарный обмен значениями

7 Выход

3

2.000000 + 3.000000i

1.000000 + 2.234000i

6.000000 + 7.000000i

5.456000 + 6.789000i

1 Создание списка

2 Вставка нового элемента в список

3 Печать списка

4 Удаление элемента из списка

5 подсчет длины списка

6 Попарный обмен значениями

7 Выход

3

2.000000 + 3.000000i

1.000000 + 2.234000i

6.000000 + 7.000000i

5.456000 + 6.789000i

1 Создание списка

2 Вставка нового элемента в список

3 Печать списка

4 Удаление элемента из списка

5 подсчет длины списка

6 Попарный обмен значениями

7 Выход

3

2.000000 + 3.000000i

1.000000 + 2.234000i

6.000000 + 7.000000i

5.456000 + 6.789000i

1 Создание списка

2 Вставка нового элемента в список

3 Печать списка

4 Удаление элемента из списка

5 подсчет длины списка

6 Попарный обмен значениями

7 Выход

4

Введите целую и мнимую часть числа для удаления элемента из узла

6

7

1 Создание списка

2 Вставка нового элемента в список

3 Печать списка

4 Удаление элемента из списка

5 подсчет длины списка

6 Попарный обмен значениями

7 Выход

3

5.456000 + 6.789000i

1.000000 + 2.234000i

6.000000 + 7.000000i

1 Создание списка

2 Вставка нового элемента в список

3 Печать списка

4 Удаление элемента из списка

5 подсчет длины списка

6 Попарный обмен значениями

7 Выход

4

Введите целую и мнимую часть числа для удаления элемента из узла

1

2.234

1 Создание списка

2 Вставка нового элемента в список

3 Печать списка

4 Удаление элемента из списка

5 подсчет длины списка

6 Попарный обмен значениями

7 Выход

3

6.000000 + 7.000000i

1.000000 + 2.234000i

1 Создание списка

2 Вставка нового элемента в список

3 Печать списка

4 Удаление элемента из списка

5 подсчет длины списка

6 Попарный обмен значениями

7 Выход

5

Количество элементов = 2

1 Создание списка

2 Вставка нового элемента в список

3 Печать списка

4 Удаление элемента из списка

5 подсчет длины списка

6 Попарный обмен значениями

7 Выход

7

1. **Дневник отладки** должен содержать дату и время сеансов отладки и основные события (ошибки в сценарии и программе, нестандартные ситуации) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Лаб. или дом. | Дата | Время | Событие | Действие по исправлению | Примечание |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

1. **Замечания автора** по существу работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 11. **Выводы**

Закрепил знания по динамическим структурам, научился реализовывать циклический односвязный список.

Подпись студента\_\_\_\_\_\_\_\_\_\_\_\_