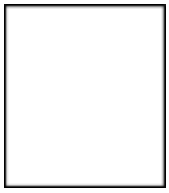

Отчёт по лабораторной работе №. 25-26 по курсу 1

Практикум на ЭВМ

студента группы М80-104б-18 Сыроежкина Кирилла Геннадьевича, №. по списку 18

Адреса www, e-mail, jabber, skype KrillsA@yandex.ru

Работа выполнена: “14“ мая 2019г.

Преподаватель: Доцент каф.806 Никулин С.П.

Входной контроль знаний с оценкой

Отчёт сдан “ “ 2019 г., итоговая оценка

Подпись преподавателя

1. **Тема**: Абстрактные типы данных. Модульное программирование на языке Си. Автоматизация сборки программ модульной структуры на языке Си с использованием утилиты make.

- **Цель работы**: Составить и отладить модуль определений и модуль реализации по заданной схеме модуля определений для абстрактного типа данных.
- **Задание** (*2*): Вставка элемента в список, упорядоченный по возрастанию, с сохранением порядка(сортировка вставкой)
- **Оборудование** (*лабораторное*):
  ЭВМ 1 , процессор Intel Celeron i686 , имя узла сети client 1 с ОП 1000 ____ МБ

НМД 70 ГБ. Терминал lxterminal адрес: 192.168.2.37
. Принтер
Другие устройства

Оборудование ПЭВМ студента, если использовалось:
Процессор Intel core i7-7700 , ОП 16384 МБ,
НМД 1024 ГБ. Монитор BENQ GW2470
Другие устройства

- **Программное обеспечение** (*лабораторное*):
  Операционная система семейства UNIX, наименование Ubuntu версия 16.04
  Интерпретатор команд bash версия
  Система программирования Си
  версия
  Редактор текстов emacs
  версия
  Утилиты операционной системы cmp ,comm, wc, dd, diff, grep, join, sort ,tail, tee, tr, uniq, od, sum

  Прикладные системы и программы gnuplot, bc

  Местонахождения и имена файлов программ и данных /std/188237

Программное обеспечение ЭВМ студента, если использовалось:
Операционная система семейства Windows , наименование Windows 10 версия 10.0.17763.316
Интерпретатор команд cmd версия
Система программирования Си версия
Редактор текстов Sublime text 3 версия 3.1.1
Утилиты операционной системы проводник
Прикладные системы и программы Yandex Browser, notepad++
Местонахождения и имена файлов программ и данных C:\Kirill

- **Идея, метод, алгоритм**

Сортировка вставками:
Первый элемент в массиве образует уже отсортированную последовательность. Сравниваем второй элемент с первым. Если порядок между ними нарушен, то первый элемент передвигается на одну позицию вправо. Теперь отсортированный массив состоит из двух элементов.
Далее, в течении каждой итерации, берем следующий элемент (третий, четвертый и т.д) и сравниваем его поочередно с другими элементами в уже отсортированном списке, начиная с конца этого списка. Если порядок между сравниваемыми элементами нарушен, то меняем их местами, если нет, то “вставка” нового элемента закончена, переходим к следующему.

ПРОГРАММА

Список на массиве реализуем по индексному доступу.

Так как в условие не указан вид списка, то пусть он будет двусвязным.
Основные функции:
list_delete_elem - удаляет элемент списка и сдвигает остальные элементы влево.
elem_sort - получает на вход элемент и вставляет его в нужное место в уже отсортированной последовательности(по сути упращенная сортировка вставками).
list_good_insert-добавляет элемент в список и функцией elem_sort вставляет его на нужное место.
+ функцией list_update обновляет длину списка и связи элементов.

Сценарий выполнения работы

Создаие функционального файла:
Создать структуру списка
Создать функции-итераторы для дальнейшей работы со списком
Создать функции для вставки/удаления элементов
Создать функцию для печати меню
Создать заголовочный файл с описанием всех функций
Создать main файл
Создать makefile

Допущен к выполнению работы. Подпись преподавателя

- **Распечатка протокола**

kircatle@DESKTOP-70J5NO3:/mnt/c/Users/joker/Desktop/LabsMAI/SecondSem/lab26$ cat **makefile**

```
# makefile 1
CC=cc # имя компилятора
LD=cc # имя редактора связей
```

```
LDFLAGS = -o # флаги для редактора связей
CCFLAGS = -c # флаги для компилятора
lab25_26: main.o list.o
	$(LD) $(LDFLAGS) lab25_26 main.o list.o
main.o: main.c list.h
	$(CC) $(CCFLAGS) main.c
list.o: list.c
	$(CC) $(CCFLAGS) list.c
cc  -o  lab25_26 main.o list.o
```

kircatle@DESKTOP-70J5NO3:/mnt/c/Users/joker/Desktop/LabsMAI/Second Sem/lab26$ cat list.c

```
#include <stdio.h>
#include <stdbool.h>
int max_size;
typedef struct
{
   int keyint;
   int num;
}t_data;
typedef struct
{
   t_data key;
   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].key.num=0;
         l[i].key.keyint=i;
      }
   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 (i==0)
      {
         l[i].next_index=i+1;
         l[i].prev_index=-1;
         l->first_index=i;
      }
      else if (i==size-1)
      {
         l[i].next_index=i+1;
         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].key.num;
```

```
}
void list_store(list* l, int i, int t, int key) // Итератор, присваивающий значение
по индексу
{
   l[i].key.num=t;
   l[i].key.keyint=key;
}
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_delete_elem(list* l, int i) //Удаление элемента из списка
{
   if (i>list_size(l))
   {
      printf("\nЭлемента с таким индексном не существует!\n");
      return false;
   }
   for (i; i<list_size(l); i++)
      list_store(l,i,list_fetch(l,list_next_elem(l,i)), i);
   list_store(l, list_size(l)-1, 0, 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_destroy(list *l) //очистить список
{
   for (int i=0; i<max_size; i++)
      list_store(l, i, 0,0);
   l->size=0;
}

void elem_sort(list* l, int newElement) //кидаем элемент на нужное место
{
   int location=list_size(l)-1;
   while(location > list_first_elem(l) && list_fetch(l, list_prev_elem(l,
location))> newElement)
   {
       list_store(l, location, list_fetch(l, list_prev_elem(l, location)),
location);
        location=list_prev_elem(l, location);
   }
   list_store(l, location, newElement, location);
}
void list_good_insert(list* l, int newElement) //вставка элемента в правильном
порядке
{
   list_update(l, list_size(l)+1);
   list_store(l,list_size(l)-1, newElement, list_size(l)-1);
   elem_sort(l, newElement);
}
void print_menu()
{
```

```
   printf("\n1.Вставить элемент в список с соблюдением порядка. \n2.Удалить элемент из списка\n3.Писка\n3.Печать списка.\n4.Длина списка.\n5.Очистить список.\nВыберите действие: ");
```
}**kircatle@DESKTOP-70J5NO3:/mnt/c/Users/joker/Desktop/LabsMAI/SecondSem/lab26$** cat **list.h**
```
#ifndef _list_h_
#define _list_h_
#include <stdio.h>
#include <stdbool.h>
int max_size; //максимальный размер спикска
typedef struct
{
   int keyint; //ключ
   int num; //значение
}t_data; //данные

typedef struct
{
   t_data key; //данные
   int next_index; // следующий элем
   int prev_index; //предыдущий элем
   int last_index; //последний элемент
   int first_index; //первый элемент
   int size; // длина

} list;

int list_create(list* l);  //Объявление списка
int list_update(list *l, int size);  //обновление длины списка
int list_next_elem(list* l, int i); //Итератор следующего элемента
int list_prev_elem(list* l, int i); // Итератор предыдущего элемента
int list_fetch(list* l, int i); // Итератор, дающий значение по индексу
void list_store(list* l, int i, int t); // Итератор, присваивающий значение по индексу
int list_first_elem(list* l); // Итератор, возвращающий первый элемент
int list_last_elem(list* l); // Итератор, возвращающий последний элемент
int list_size(list* l); //Итератор, возвращающий длину списка
bool list_empty(list* l); //Итератор, проверяющий на пустоту список
bool list_delete_elem(list* l, int i); //Удаление элемента из списка
```

```
void list_print(list* l); //Печать списка
void list_destroy(list *l); //очистить список
void elem_sort(list* l, int newElement); // кидает полученный элемент на
нужное место(сортировка вставками)
void list_good_insert(list* mass, int newElement); //Вставка элемента в список в
правильном порядке
void print_menu(); // печать
```
меню**kircatle@DESKTOP-70J5NO3:/mnt/c/Users/joker/Desktop/LabsMAI/SecondSem/lab26$** cat **main.c**
```
#include <stdio.h>
#include <stdbool.h>
#include "list.h"
int main()
{
   printf("Введите максимальный размер линейного 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_good_insert(l, elem);
         break;
         case 2:
            printf("\nВыберите на какой позици удалить элемент(от 1): ");
            scanf("%d", &position);
            list_delete_elem(l, position-1);
         break;
         case 3:
            list_print(l);
         break;
         case 4:
            printf("\n%d\n",list_size(l));
```

```
            break;
            case 5:
                list_destroy(l);
            break;
        }
         print_menu();
    }
    printf("\n");
```

}**kircatle@DESKTOP-70J5NO3:/mnt/c/Users/joker/Desktop/LabsMAI/SecondSem/lab26$** make
cc -c main.c
cc -c list.c
cc -o lab25_26 main.o list.o

Тесты

kircatle@DESKTOP-70J5NO3:/mnt/c/Users/joker/Desktop/LabsMAI/SecondSem/lab26$ ./lab25_26
Введите максимальный размер линейного списка: 6

1.Вставить элемент в список с соблюдением порядка.
2.Удалить элемент из списка
3.Печать списка.
4.Длина списка.
5.Очистить список.
Выберите действие: 1

Элемент: 999

1.Вставить элемент в список с соблюдением порядка.
2.Удалить элемент из списка
3.Печать списка.
4.Длина списка.
5.Очистить список.
Выберите действие: 1

Элемент: 5

1.Вставить элемент в список с соблюдением порядка.

2.Удалить элемент из списка
3.Печать списка.
4.Длина списка.
5.Очистить список.
Выберите действие: 1

Элемент: 9

1.Вставить элемент в список с соблюдением порядка.
2.Удалить элемент из списка
3.Печать списка.
4.Длина списка.
5.Очистить список.
Выберите действие: 1

Элемент: 6

1.Вставить элемент в список с соблюдением порядка.
2.Удалить элемент из списка
3.Печать списка.
4.Длина списка.
5.Очистить список.
Выберите действие: 3

5 6 9 999

1.Вставить элемент в список с соблюдением порядка.
2.Удалить элемент из списка
3.Печать списка.
4.Длина списка.
5.Очистить список.
Выберите действие: 1

Элемент: 8

1.Вставить элемент в список с соблюдением порядка.
2.Удалить элемент из списка
3.Печать списка.
4.Длина списка.

5.Очистить список.
Выберите действие: 1

Элемент: 3

1.Вставить элемент в список с соблюдением порядка.
2.Удалить элемент из списка
3.Печать списка.
4.Длина списка.
5.Очистить список.
Выберите действие: 3

3 5 6 8 9 999

1.Вставить элемент в список с соблюдением порядка.
2.Удалить элемент из списка
3.Печать списка.
4.Длина списка.
5.Очистить список.
Выберите действие: 2

Выберите на какой позици удалить элемент(от 1): 4

1.Вставить элемент в список с соблюдением порядка.
2.Удалить элемент из списка
3.Печать списка.
4.Длина списка.
5.Очистить список.
Выберите действие: 3

3 5 6 9 999

1.Вставить элемент в список с соблюдением порядка.
2.Удалить элемент из списка
3.Печать списка.
4.Длина списка.
5.Очистить список.
Выберите действие: 4

5

1.Вставить элемент в список с соблюдением порядка.
2.Удалить элемент из списка
3.Печать списка.
4.Длина списка.
5.Очистить список.
Выберите действие: 1

Элемент: 7

1.Вставить элемент в список с соблюдением порядка.
2.Удалить элемент из списка
3.Печать списка.
4.Длина списка.
5.Очистить список.
Выберите действие: 3

3 5 6 7 9 999

1.Вставить элемент в список с соблюдением порядка.
2.Удалить элемент из списка
3.Печать списка.
4.Длина списка.
5.Очистить список.
Выберите действие: 5

1.Вставить элемент в список с соблюдением порядка.
2.Удалить элемент из списка
3.Печать списка.
4.Длина списка.
5.Очистить список.
Выберите действие: 3

1.Вставить элемент в список с соблюдением порядка.
2.Удалить элемент из списка
3.Печать списка.

4.Длина списка.
5.Очистить список.
Выберите действие: EOF
kircatle@DESKTOP-70J5NO3:/mnt/c/Users/joker/Desktop/LabsMAI/Second Sem/lab26$ ./lab25_26
Введите максимальный размер линейного списка: 9

1.Вставить элемент в список с соблюдением порядка.
2.Удалить элемент из списка
3.Печать списка.
4.Длина списка.
5.Очистить список.
Выберите действие: 1

Элемент: 99

1.Вставить элемент в список с соблюдением порядка.
2.Удалить элемент из списка
3.Печать списка.
4.Длина списка.
5.Очистить список.
Выберите действие: 1

Элемент: 3

1.Вставить элемент в список с соблюдением порядка.
2.Удалить элемент из списка
3.Печать списка.
4.Длина списка.
5.Очистить список.
Выберите действие: 1

Элемент: 2

1.Вставить элемент в список с соблюдением порядка.
2.Удалить элемент из списка
3.Печать списка.
4.Длина списка.
5.Очистить список.

Выберите действие: 1

Элемент: 8888

1.Вставить элемент в список с соблюдением порядка.
2.Удалить элемент из списка
3.Печать списка.
4.Длина списка.
5.Очистить список.
Выберите действие: 1

Элемент: 99

1.Вставить элемент в список с соблюдением порядка.
2.Удалить элемент из списка
3.Печать списка.
4.Длина списка.
5.Очистить список.
Выберите действие: 1

Элемент: 4

1.Вставить элемент в список с соблюдением порядка.
2.Удалить элемент из списка
3.Печать списка.
4.Длина списка.
5.Очистить список.
Выберите действие: 3

2 3 4 99 99 8888

1.Вставить элемент в список с соблюдением порядка.
2.Удалить элемент из списка
3.Печать списка.
4.Длина списка.
5.Очистить список.
Выберите действие: 2

Выберите на какой позици удалить элемент(от 1): 6

1.Вставить элемент в список с соблюдением порядка.
2.Удалить элемент из списка
3.Печать списка.
4.Длина списка.
5.Очистить список.
Выберите действие: 3

2 3 4 99 99

1.Вставить элемент в список с соблюдением порядка.
2.Удалить элемент из списка
3.Печать списка.
4.Длина списка.
5.Очистить список.
Выберите действие: 4

5

1.Вставить элемент в список с соблюдением порядка.
2.Удалить элемент из списка
3.Печать списка.
4.Длина списка.
5.Очистить список.
Выберите действие:

- **Дневник отладки**

№.	Лаб. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание
1	дом	10.5.13	19:00	Не во всех случаях работала сортировка вставками	Внимательно ещё раз просмотрел работу алгорима. Проблема была устранена	

- Замечание автора по существу работы теперь я владею навыком модульного программирования
- Выводы Я составил модульную программу
  устранены следующим образом Внимательное прочтение алгоритмов.

Подпись студента