



Отчёт по лабораторной работе № 25-26 по курсу 1

студента группы M80-108Б-19 Хренниковой Ангелины, № по списку 23

Адреса www, e-mail, jabber, skype: lina.khrennikova@mail.ru

Работа выполнена: “25” марта 2020г.

Преподаватель: Поповкин А. В. каф.806

Входной контроль знаний с оценкой

Отчёт сдан “26” марта 2020 г., итоговая оценка

Подпись преподавателя

1. **Тема:** Абстрактные типы данных. Рекурсия. Модульное программирование на языке Си. Автоматизация сборки программ модульной структуры на языке Си с использованием утилиты make.
2. **Цель работы:** 1. Изучить принципы работы утилиты make по книгам. 2. Составить Makefile для модульной программы(составить и отладить определений и модуль реализации по заданной схеме определений для абстрактного типа данных; составить программный модуль, сортирующий экземпляр указанного абстрактного типа данных заданным методом, используя только операции, импортированные из модуля UUDT). 3. Оттестировать Makefile и убедиться в его работоспособности для различных ситуаций. 4. Распечатать протокол с текстом Makefile и результат его работы в нескольких различных ситуациях с исходными и объектными файлами. В протокол включить должным образом атрибутированные списки файлов, документирующие состояние файловой системы проекта.

3. **Задание (вариант № ):** Составить и отладить определений и модуль реализации по заданной схеме определений(вставка элемента в дек, упорядоченного по возрастанию, с сохранением порядка) для абстрактного типа данных(дек); составить программный модуль, сортирующий экземпляр указанного абстрактного типа данных заданным методом(сортировка простоой вставкой), используя только операции, импортированные из модуля UUDT

4. **Оборудование (лабораторное):**  
ЭВМ PC, процессор Intel® Core™ i7-3770 CPU @ 3.40GHz \* 8, имя узла сети alise18, с ОП 15974.4 МБ, НМД 345.5 ГБ.  
Терминал Gnome адрес 192.168.2.118/24. Принтер  
Другие устройства

Оборудование ПЭВМ студента, если использовалось:

Процессор Intel® Core™ i3-7020U CPU @ 2.30GHz \* 4, ОП 8192 МБ, НМД 256 ГБ. Монитор LCD  
Другие устройства

5. **Программное обеспечение (лабораторное):**  
Операционная система семейства UNIX, наименование Ubuntu версия 18.04  
Интерпретатор команд Bash версия 4.4.20(1)  
Система программирования версия  
Редактор текстов Nano версия 2.9.3  
Утилиты операционной системы touch, make, cat, ls  
Прикладные системы и программы  
Местонахождения и имена файлов программ и данных home/stud/olen

Программное обеспечение ЭВМ студента, если использовалось:

Операционная система семейства UNIX, наименование Ubuntu версия 18.04

Интерпретатор команд	<u>Bash</u>	версия	<u>4.4.19(1)</u>
Система программирования		версия	
Редактор текстов	<u>Emacs</u>	версия	<u>25.2.2</u>
Утилиты операционной системы	<u>touch, cat, ls, make</u>		
<hr/>			
Прикладные системы и программы			
Местонахождения и имена файлов программ и данных	<u>home/lina_tucha</u>		
<hr/>			

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

Сортировка вставками — это простой алгоритм сортировки. Суть его заключается в том что, на каждом шаге алгоритма мы берем один из элементов массива, находим позицию для вставки и вставляем. Стоит отметить что массив из 1-го элемента считается отсортированным.

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

На вход алгоритма подаётся последовательность  $n$  чисел:  $a_1, a_2, \dots, a_n$ . Сортируемые числа также называют ключами. Входная последовательность на практике представляется в виде массива с  $n$  элементами. На выходе алгоритм должен вернуть перестановку исходной последовательности  $a_1', a_2', \dots, a_n'$ , чтобы выполнялось следующее соотношение  $a_1' \leq a_2' \leq \dots \leq a_n'$ .

В начальный момент отсортированная последовательность пуста. На каждом шаге алгоритма выбирается один из элементов входных данных и помещается на нужную позицию в уже отсортированной последовательности до тех пор, пока набор входных данных не будет исчерпан. В любой момент времени в отсортированной последовательности элементы удовлетворяют требованиям к выходным данным алгоритма.

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

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

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

```
lina_tucha@LAPTOP-44CRFC1U:~$ cat MF
#Makefile for lab26
CC = gcc
LD = gcc
CCFLAGS = -g -Wall -std=c99
OBJ = lab26.o deque.o sort.o
lab26.out: $(OBJ)
    $(LD) -o lab26.out $(OBJ)
lab26.o: lab26.c deque.h sort.h
    $(CC) $(CCFLAGS) -c lab26.c
deque.o: deque.c deque.h
    $(CC) $(CCFLAGS) -c deque.c
sort.o: sort.c sort.h deque.h
    $(CC) $(CCFLAGS) -c sort.c
clean:
    rm -f *.o
lina_tucha@LAPTOP-44CRFC1U:~$ cat deque.h
#ifndef _DEQUE_H_
#define _DEQUE_H_
#include <stdbool.h>
typedef struct{
    int key;
    double value;
}Item;
typedef struct{
    int front, back, size;
    Item data[10];
}Deque;
void d_create(Deque*);
bool d_empty(const Deque*);
```

```

int d_size(const Deque*);
void d_push_front(Deque*,Item);
void d_push_back(Deque*,Item);
Item d_pop_front(Deque*);
Item d_pop_back(Deque*);
void d_print(const Deque*);
#endif
lina_tucha@LAPTOP-44CRFC1U:~$ cat deque.c
#include <stdlib.h>
#include <stdio.h>
#include "deque.h"
void d_create(Deque* d){d->size=0;}
bool d_empty(const Deque* d){
    return d->size==0;
}
int d_size(const Deque* d){return d->size;}
void d_push_front(Deque* d, Item item){
    if(d->size<10){
        if(d->size==0){
            d->front=0;
            d->back=0;
        }
        else{
            d->front=((d->front-1)%10+10)%10;
        }
        d->data[d->front]=item;
        d->size++;
    }
    else{printf("Haven't free place\n");}
}
void d_push_back(Deque* d, Item item){
    if(d->size<10){
        if(d->size==0){
            d->front=0;
            d->back=0;
        }
        else{
            d->back=(d->back+1)%10;
        }
        d->data[d->back]=item;
        d->size++;
    }
    else{printf("Haven't free place\n");}
}
Item d_pop_front(Deque* d){
    Item res;
    if(!d_empty(d)){
        res = d->data[d->front];
        d->front=(d->front+1)%10;
        d->size--;
    }return res;
}
Item d_pop_back(Deque* d){
    Item res;
    if(!d_empty(d)){
        res = d->data[d->back];
        d->back=((d->back-1)%10+10)%10;
        d->size--;
    }return res;
}
void d_print(const Deque* d){
    int i=d->front;
    int s=d->size;
    printf("<");
    while(s>0){
        printf("(%d, %.2lf)",d->data[i].key,d->data[i].value);
        i=(i+1)%10;
        s--;
    }
    printf(">\n");
}
lina_tucha@LAPTOP-44CRFC1U:~$ cat sort.h

```

```

#ifndef _SORT_H_
#define _SORT_H_
Deque concatenateDeque(Deque*,Deque*);
void quicksort(Deque*);
#endif
lina_tucha@LAPTOP-44CRFC1U:~$ cat sort.c
#include <stdbool.h>
#include <stdio.h>
#include "deque.h"
#include "sort.h"
Deque concatenate(Deque* d1, Deque* d2){
    Deque res;
    d_create(&res);
    Item I;
    while(!d_empty(d1)){
        I=d_pop_front(d1);
        d_push_back(&res,I);
    }
    while(!d_empty(d2)){
        I=d_pop_front(d2);
        d_push_back(&res,I);
    }
    return res;
}
void quicksort(Deque* d){
    if(!d_empty(d)){
        Deque less, more;
        d_create(&less);
        d_create(&more);
        Item I=d_pop_front(d);
        while(!d_empty(d)){
            Ip=d_pop_front(d);
            if(Ip.key<I.key){
                d_push_back(&less,Ip);
            }
            else{
                d_push_back(&more,Ip);
            }
        }
        quicksort(&less);
        quicksort(&more);
        d_push_back(d,I);
        *d=concatenate(&less,d);
        *d=concatenate(d,&more);
    }
}
lina_tucha@LAPTOP-44CRFC1U:~$ cat lab26.c
#include "deque.h"
#include "sort.h"
#include <stdlib.h>
#include <stdio.h>
void menu(){
    printf("\n0'-exit      '3'-sort      '6'-delete from the left\n");
    printf("\n1'-print deque  '4'-add at the left  '7'-delete from the right\n");
    printf("\n2'-size      '5'-add at the right\n");
}
int main(){
    menu();
    Deque de;
    d_create(&de);
    char command='#';
    Item I;
    while(command!='0'){
        scanf("%c",&command);
        fflush(stdin);
        switch (command){
            case '0':break;
            case '1':
                d_print(&de);
                menu();
            break;
            case '2':

```

```

        printf("Size=%d\n",d_size(&de));
        menu();
    break;
    case '3':
        quicksort(&de);
        menu();
    break;
    case '4':
        printf("Enter pair <key> <value>: ");
        scanf("%d%lf",&I.key,&I.value);
        d_push_front(&de,I);
        menu();
    break;
    case '5':
        printf("Enter pair <key> <value>: ");
        scanf("%d%lf",&I.key,&I.value);
        d_push_back(&de,I);
        menu();
    break;
    case '6':
        d_pop_front(&de);
        menu();
    break;
    case '7':
        d_pop_back(&de);
        menu();
    break;
    case ' ':break;
    case '\n':break;
    default:
        printf("Such command is not in menu.\n");
    break;
}

}

lina_tucha@LAPTOP-44CRFC1U:~$
lina_tucha@LAPTOP-44CRFC1U:~$ make -f MF
gcc -g -Wall -std=c99 -c lab26.c
gcc -g -Wall -std=c99 -c deque.c
gcc -g -Wall -std=c99 -c sort.c
gcc -o lab26.out lab26.o deque.o sort.o
lina_tucha@LAPTOP-44CRFC1U:~$ make -f MF
make: «lab26.out» не требует обновления.
lina_tucha@LAPTOP-44CRFC1U:~$ ls -l|tr -s ' '|cut -d' ' -f6- |head -11|sort -r
map 26 19:34 sort.o
map 26 19:34 lab26.out
map 26 19:34 lab26.o
map 26 19:34 deque.o
map 26 19:27 deque.c
map 26 00:17 MF
map 26 00:16 lab26.c
map 25 22:34 deque.h
map 25 21:56 sort.c
map 25 18:37 sort.h
lina_tucha@LAPTOP-44CRFC1U:~$ lab26.o
lina_tucha@LAPTOP-44CRFC1U:~$ make -f MF
gcc -o lab26.out lab26.o deque.o sort.o
lina_tucha@LAPTOP-44CRFC1U:~$ ls -l|tr -s ' '|cut -d' ' -f6- |head -11|sort -r
map 26 19:35 lab26.out
map 26 19:35 lab26.o
map 26 19:34 sort.o
map 26 19:34 deque.o
map 26 19:27 deque.c
map 26 00:17 MF
map 26 00:16 lab26.c
map 25 22:34 deque.h
map 25 21:56 sort.c
map 25 18:37 sort.h
lina_tucha@LAPTOP-44CRFC1U:~$ touch sort.c
lina_tucha@LAPTOP-44CRFC1U:~$ ls -l|tr -s ' '|cut -d' ' -f6- |head -11|sort -r
map 26 19:36 sort.c
map 26 19:35 lab26.out

```

```

map 26 19:35 lab26.o
map 26 19:34 sort.o
map 26 19:34 deque.o
map 26 19:27 deque.c
map 26 00:17 MF
map 26 00:16 lab26.c
map 25 22:34 deque.h
map 25 18:37 sort.h
lina_tucha@LAPTOP-44CRFC1U:~$ make -f MF
gcc -g -Wall -std=c99 -c sort.c
gcc -o lab26.out lab26.o deque.o sort.o
lina_tucha@LAPTOP-44CRFC1U:~$ ls -l|tr -s ' ' |cut -d' ' -f6- |head -11|sort -r
map 26 19:36 sort.o
map 26 19:36 sort.c
map 26 19:36 lab26.out
map 26 19:35 lab26.o
map 26 19:34 deque.o
map 26 19:27 deque.c
map 26 00:17 MF
map 26 00:16 lab26.c
map 25 22:34 deque.h
map 25 18:37 sort.h
lina_tucha@LAPTOP-44CRFC1U:~$ touch lab26.c sort.c deque.c
lina_tucha@LAPTOP-44CRFC1U:~$ ls -l|tr -s ' ' |cut -d' ' -f6- |head -11|sort -r
map 26 19:37 sort.c
map 26 19:37 lab26.c
map 26 19:37 deque.c
map 26 19:36 sort.o
map 26 19:36 lab26.out
map 26 19:35 lab26.o
map 26 19:34 deque.o
map 26 00:17 MF
map 25 22:34 deque.h
map 25 18:37 sort.h
lina_tucha@LAPTOP-44CRFC1U:~$ make -f MF
gcc -g -Wall -std=c99 -c lab26.c
gcc -g -Wall -std=c99 -c deque.c
gcc -g -Wall -std=c99 -c sort.c
gcc -o lab26.out lab26.o deque.o sort.o
lina_tucha@LAPTOP-44CRFC1U:~$ touch deque.h
lina_tucha@LAPTOP-44CRFC1U:~$ ls -l|tr -s ' ' |cut -d' ' -f6- |head -11|sort -r
map 26 19:38 deque.h
map 26 19:37 sort.o
map 26 19:37 sort.c
map 26 19:37 lab26.out
map 26 19:37 lab26.o
map 26 19:37 lab26.c
map 26 19:37 deque.o
map 26 19:37 deque.c
map 26 00:17 MF
map 26 18:37 sort.h
lina_tucha@LAPTOP-44CRFC1U:~$ make -f MF
gcc -g -Wall -std=c99 -c lab26.c
gcc -g -Wall -std=c99 -c deque.c
gcc -g -Wall -std=c99 -c sort.c
gcc -o lab26.out lab26.o deque.o sort.o
lina_tucha@LAPTOP-44CRFC1U:~$ ls -l|tr -s ' ' |cut -d' ' -f6- |head -11|sort -r
map 26 19:38 sort.o
map 26 19:38 lab26.out
map 26 19:38 lab26.o
map 26 19:38 deque.o
map 26 19:38 deque.h
map 26 19:37 sort.c
map 26 19:37 lab26.c
map 26 19:37 deque.c
map 26 00:17 MF
map 25 18:37 sort.h
lina_tucha@LAPTOP-44CRFC1U:~$ make -f MF clean
rm -f *.o
lina_tucha@LAPTOP-44CRFC1U:~$ ls -l|tr -s ' ' |cut -d' ' -f6- |sort -r
авг 25 2019 Шаблоны
авг 25 2019 Общедоступные

```

```

авг 25 2019 Музыка
авг 25 2019 Изображения
авг 25 2019 Видео
map 26 19:38 lab26.out
map 26 19:38 deque.h
map 26 19:37 sort.c
map 26 19:37 lab26.c
map 26 19:37 deque.c
map 26 00:17 MF
map 22 22:00 Документы
map 26 18:37 sort.h
map 2 23:27 Загрузки
map 2 03:02 Рабочий стол
lina_tucha@LAPTOP-44CRFC1U:~$
lina_tucha@LAPTOP-44CRFC1U:~$ ./lab26.out
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
1
<>
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
3
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
1
<>
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
2
Size=0
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
5
Enter pair <key> <value>: 2 2.45
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
1
<(2, 2.45)>
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
3
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
1
<(2, 2.45)>
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
2
Size=1
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
4
Enter pair <key> <value>: 4 4.2
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
4
Enter pair <key> <value>: 4 4.8
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
1

```

```

<(4, 4.80)(4, 4.20)(2, 2.45)>
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
5
Enter pair <key> <value>: 3 3.15
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
5
Enter pair <key> <value>: 3 3
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
1
<(4, 4.80)(4, 4.20)(2, 2.45)(3, 3.15)(3, 3.00)>
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
5
Enter pair <key> <value>: 1 1.2
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
5
Enter pair <key> <value>: 1 1
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
1
<(4, 4.80)(4, 4.20)(2, 2.45)(3, 3.15)(3, 3.00)(1, 1.20)(1, 1.00)>
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
2
Size=7
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
3
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
1
<(1, 1.20)(1, 1.00)(2, 2.45)(3, 3.15)(3, 3.00)(4, 4.80)(4, 4.20)>
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
6
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
7
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
7
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
7
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
7
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
1
<(1, 1.00)(2, 2.45)>
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right

```



```

2'-size      '5'-add at the right
7
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
1
<(1, 1.00)>
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
4
Enter pair <key> <value>: 2 0
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
5
Enter pair <key> <value>: 0 1
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
5
Enter pair <key> <value>: -1 1
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
1
<(2, 0.00)(1, 1.00)(0, 1.00)(-1, 1.00)>
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
3
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
1
<(-1, 1.00)(0, 1.00)(1, 1.00)(2, 0.00)>
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
7
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
7
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
7
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
1
<(-1, 1.00)>
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
7
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
1
<>
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
5
Enter pair <key> <value>: -1 0
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
5
Enter pair <key> <value>: -2 0

```

```

'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
5
Enter pair <key> <value>: -3 0
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
5
Enter pair <key> <value>: -4 0
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
5
Enter pair <key> <value>: -5 0
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
5
Enter pair <key> <value>: -6 0
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
5
Enter pair <key> <value>: -7 0
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
5
Enter pair <key> <value>: -8 0
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
5
Enter pair <key> <value>: -9 0
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
2
Size=9
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
5
Enter pair <key> <value>: -10 0
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
2
Size=10
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
5
Enter pair <key> <value>: 0 0
Haven't free place
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
4
Enter pair <key> <value>: 0 0
Haven't free place
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
1
<(-1, 0.00)(-2, 0.00)(-3, 0.00)(-4, 0.00)(-5, 0.00)(-6, 0.00)(-7, 0.00)(-8, 0.00)(-9, 0.00)(-10, 0.00)>
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
3
'0'-exit      '3'-sort      '6'-delete from the left

```

```
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
1
<(-10, 0.00)(-9, 0.00)(-8, 0.00)(-7, 0.00)(-6, 0.00)(-5, 0.00)(-4, 0.00)(-3, 0.00)(-2, 0.00)(-1, 0.00)>
'0'-exit      '3'-sort      '6'-delete from the left
'1'-print deque '4'-add at the left '7'-delete from the right
'2'-size      '5'-add at the right
0
```

9. **Дневник отладки** должен содержать дату и время сеансов отладки, и основные ошибки (ошибки в сценарии и программе, не стандартные операции) 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

10. Замечание автора по существу работы \_\_\_\_\_

11. Выводы : Я научилась составлять и отлаживать определений и модуль реализации по заданной схеме определений(вставка элемента в дек, упорядоченного по возрастанию, с сохранением порядка) для абстрактного типа данных(дек); составлять программный модуль, сортирующий экземпляр указанного абстрактного типа данных заданным методом(сортировка простоой вставкой), используя только операции, импортированные из модуля UUDT

Недочеты, допущенные при выполнении задания, могут быть устранены следующим образом \_\_\_\_\_

Подпись студента Хренникова А. С.