Отчет по лабораторной работе №25-26 по курсу практикум на ЭВМ

Студент группы М8О-102Б-21 Кажекин Денис, № по списку 9

Контакты www, e-mail, icq, skype deniskazhekin@mail.ru

Работа выполнена: « » _____202___г.

Преподаватель: доцент каф. 806 Никулин Сергей Петрович

Входной контроль знаний с оценкой _____

Отчет сдан « » _____202 __ г., итоговая оценка _____

Подпись преподавателя _____

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

2. **Цель работы:** Изучить утилиту make, абстрактные типы данных, модульное программирование и рекурсию.

3. **Задание** (*вариант №* 3;5 **):**

   3 - Дек.

   5 - Процедура: слияние двух деков, упорядоченных по возрастанию, с сохранением порядка.
   Метод: сортировка слиянием.

4. **Оборудование** (лабораторное):
   Оборудование ПЭВМ студента, если использовалось:
   Процессор: Apple M1, с ОП 8192 Мб, НМД 262144 Мб. Монитор: Retina 13,3; IPS 2560 x 1600.
5. **Программное обеспечение (лабораторное):**
   Операционная система семейства Linux, наименование Ubuntu, версия ____20.04.3 LTS____интерпретатор команд _bash_ версия __5.0.17(1)__

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

Относительно 25 лабораторной работы:

1)Сконструируем makefile

Относительно 26 лабораторной работы:

1) Реализуем модуль дека и методы отдельно

2)Пользователь будет взаимодействовать с программой благодаря меню, которое содержит следующие функции:

1. Создать дек 2. Проверить дек на пустоту 3. Печать размера дека 4. Добавить элемент в начало дека 5. Добавить элемент в конец дека 6. Top с начала дека 7. Top с конца дек 8. Удалить элемент в начале дека 9. Удалить элемент в конце дек 10. Распечатать весь дек 11. Сортировать дек 12. Выход

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

Произвольно вводим команды, тестируя программу

Пункты 1-7 отчета составляются строго до начала лабораторной работы.
Допущен к выполнению работы. **Подпись преподавателя** _____
_____

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

```
deniskazhekin@MacBook-Air-Denis ~ %cat deq.c
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include "deq.h"
void create_udt(udt *d, const int cap) {
	int i;
	UDT_TYPE item;
	item.key=0;
	item.value=0.0f;
	if (cap<=0) return;
	d->t=(UDT_TYPE *)malloc(sizeof(UDT_TYPE) * cap);
	for (i=0; i<cap; i++)
		d->t[i]=item;
		d->cap=cap;
		d->size=0;
		d->first=(cap == 1 ? 0 : 1);
		d->last=0;
}
int udt_empty(const udt *d) {
	return d->size==0;
}
int udt_size(const udt *d) {
	return d->size;
}
int udt_push_back(udt *d, UDT_TYPE a) {
	int p=(d->last+d->cap+1) % d->cap;
	if (d->size==d->cap) return 0;
	d->t[p]=a;
	d->last=p;
	d->size++;
	return 1;
}
int udt_push_front(udt *d, UDT_TYPE a) {
	int p=(d->first+d->cap-1) % d->cap;
	if (d->size==d->cap) return 0;
	d->t[p]=a;
	d->first=p;
	d->size++;
	return 1;
}
bool udt_pop_front(udt *d) {
	int p=(d->first+d->cap+1)% d->cap;
	UDT_TYPE item;
	item.key=0;
	item.value=0.0f;
	if (d->size==0) return false;
	d->t[d->first]=item;
	d->first=p;
	d->size--;
	return true;
}
bool udt_pop_back(udt *d) {
	int p=(d->last+d->cap+1)% d->cap;
	UDT_TYPE item;
	item.key=0;
	item.value=0.0f;
	if (d->size==0) return false;
	d->t[d->last]=item;
	d->last=p;
	d->size--;
	return true;
}
data udt_top_left(udt *d) {
	return d->t[d->first];
```

```
}
data udt_top_right(udt *d) {
	return d->t[d->last];
}
void udt_print(udt *d) {
	printf("Key\tValue\n");
	int size=udt_size(d);
	for (int i=0; i<size; i++) {
		UDT_TYPE a=d->t[(i+d->first)% d->cap];
		printf("%d\t", a.key);
		printf("%f\n", a.value);
	}
}
void udt_destroy(udt *d) {
	if (d->t != NULL) {
		free(d->t);
		d->t=NULL;
	}
	d->cap=0;
	d->size=0;
	d->first=0;
	d->last=0;
}
deniskazhekin@MacBook-Air-Denis ~ %cat deq.h
#ifndef _UDT_H_
#define _UDT_H_

#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>

typedef struct data {
   int key;
   float value;
} data;

typedef data UDT_TYPE;

typedef struct {
   int first;
   int last;
   int size;
   int cap;
   UDT_TYPE *t;
} udt;

void create_udt(udt *, const int cap);

int udt_empty(const udt *);

int udt_push_front(udt *, UDT_TYPE a);

int udt_push_back(udt *, UDT_TYPE a);

bool udt_pop_front(udt *);

bool udt_pop_back(udt *);

data udt_top_left(udt *);

data udt_top_right(udt *);

void udt_print(udt *);
```

```
int udt_size(const udt *);
void udt_destroy(udt *);

#endif
deniskazhekin@MacBook-Air-Denis ~ %cat main.c
#include <stdio.h>
#include "deq.h"
void sl(udt *d, udt *b, udt *c) {
	int size=udt_size(d);
	int i;
	for (i=0; i<(size/2); i++) {
		udt_push_back(b, udt_top_left(d));
		udt_pop_front(d);
	}
	for (; i<size; i++) {
		udt_push_back(c, udt_top_left(d));
		udt_pop_front(d);
	}
}
void proc(udt *d, udt *b, udt *c) {
	while (!udt_empty(b) && !udt_empty(c)) {
		if (udt_top_left(b).key<=udt_top_left(c).key) {
			udt_push_back(d, udt_top_left(b));
			udt_pop_front(b);
		}else{
			udt_push_back(d, udt_top_left(c));
			udt_pop_front(c);
		}
	}
	while (!udt_empty(b)) {
		udt_push_back(d, udt_top_left(b));
		udt_pop_front(b);
	}
	while (!udt_empty(c)) {
		udt_push_back(d, udt_top_left(c));
		udt_pop_front(c);
	}
}
void sort(udt *d) {
	int size=udt_size(d);
	udt l, r;
	if (size<=1) return;
	create_udt(&l, size);
	create_udt(&r, size);
	sl(d, &l, &r);
	sort(&l);
	sort(&r);
	proc(d, &l, &r);
	udt_destroy(&l);
	udt_destroy(&r);
}
int main() {
	int c=1, h;
	int n=20;
	UDT_TYPE v;
	udt *q=(udt *) malloc(sizeof(udt));
	while (c) {
		printf("1. Create deq\t 2. Is empty\t 3. Size of the deq\t 4. Add element to start\t 5. Add element to the
end\t 6. First element\t 7. Last element\t 8. Delete the first element\t 9. Delete the last element\t 10. Print the deq\t 11.
Sort\t 12. Exit\n");
		scanf("%d", &h);
		switch (h) {
```

```
case 1: {
	create_udt(q, n);
	break;
	}
case 2: {
	if (!udt_empty(q)) {
		printf("Deq is not empty\n");
	}else{
		printf("Deq is empty\n");
	}
	break;
	}
case 3: {
	printf("%d\n", udt_size(q));
	break;
	}
case 4: {
	if (q==NULL) {
		printf("There is no deq\n");
	}else{
		printf("Enter the key: ");
		scanf("%d", &v.key);
		printf("Enter the value: ");
		scanf("%f", &v.value);
		if (!udt_push_front(q, v)) {
			printf("Deq is full\n");
		}
	}
	break;
	}
case 5: {
	if (q==NULL) {
		printf("There is no deq\n");
	}else{
		printf("Enter the key: ");
		scanf("%d", &v.key);
		printf("Enter the value: ");
		scanf("%f", &v.value);
		if (!udt_push_back(q, v)) {
			printf("Deq is full\n");
		}
	}
	break;
	}
case 6: {
	if (q==NULL) {
		printf("There is no deq\n");
	}else{
		if (udt_empty(q)==1) {
			printf("Deq is empty\n");
		}else{
			data k=udt_top_left(q);
			printf("Key\n%d\nValue\n%f\n", k.key, k.value);
		}
	}
	break;
	}
case 7: {
	if (q==NULL) {
		printf("There is no deq\n");
	}else{
		if (udt_size(q)==0) {
```

```
                                        printf("Deq is empty\n");
                                }else{
                                        data s=udt_top_right(q);
                                        printf("Key\n%d\nValue\n%f\n", s.key, s.value);
                                }
                        }
                        break;
                        }
                case 8: {
                        if (q==NULL) {
                                printf("There is no deq\n");
                        }else{
                                if (!udt_pop_front(q)==1) {
                                        printf("Deq is empty\n");
                                }
                        }
                        break;
                        }
                case 9: {
                        if (q==NULL) {
                                printf("There is no deq\n");
                        }else{
                                if (!udt_pop_back(q)) {
                                        printf("Deq is empty\n");
                                }
                        }
                        break;
                        }
                case 10: {
                        if (q==NULL) {
                                printf("There is no deq\n");
                        }else{
                                udt_print(q);
                        }
                        break;
                        }
                case 11: {
                        if (q==NULL) {
                                printf("There is no deq\n");
                        }else{
                                sort(q);
                        }
                        break;
                        }
                case 12: {
                        c=0;
                        break;
                        }
                default: {
                        printf("Incorrect input\n");
                }
                }
        }
        return 0;
}
deniskazhekin@MacBook-Air-Denis ~ %cat Makefile
laba: deq.o main.o
        gcc deq.o main.o
deq.o: deq.h deq.c
        gcc -c deq.c
main.o: deq.h main.c
        gcc -c main.c
```

```
clean:
	rm -f *.o
deniskazhekin@MacBook-Air-Denis ~ % ls
Applications  Movies		PycharmProjects	untitled1	untitled6
Desktop		Music		deq.c		untitled2	untitled7
Documents	Pictures	deq.h		untitled3
Downloads	Projects	main.c		untitled4
Library		Public		untitled	untitled5
deniskazhekin@MacBook-Air-Denis ~ % make main.o
cc    -c -o main.o main.c
deniskazhekin@MacBook-Air-Denis ~ % make deq.o
cc    -c -o deq.o deq.c
deniskazhekin@MacBook-Air-Denis ~ % ls
Applications  Movies		PycharmProjects	main.o		untitled4
Desktop		Music		deq.c		untitled	untitled5
Documents	Pictures	deq.h		untitled1	untitled6
Downloads	Projects	deq.o		untitled2	untitled7
Library		Public		main.c		untitled3
deniskazhekin@MacBook-Air-Denis ~ % make clean
rm -f *.o
deniskazhekin@MacBook-Air-Denis ~ %make
gcc -c deq.c
gcc -c main.c
gcc deq.o main.o
deniskazhekin@MacBook-Air-Denis ~ %./a.out
1. Create deq  2. Is empty    3. Size of the deq     4. Add element to star 5. Add element to the end  6. First element   7.
Last element   8. Delete the first element   9. Delete the last element    10. Print the deq     11. Sort  12. Exit
1
1. Create deq  2. Is empty    3. Size of the deq     4. Add element to star 5. Add element to the end  6. First element   7.
Last element   8. Delete the first element   9. Delete the last element    10. Print the deq     11. Sort  12. Exit
2
Deq is empty
1. Create deq  2. Is empty    3. Size of the deq     4. Add element to star 5. Add element to the end  6. First element   7.
Last element   8. Delete the first element   9. Delete the last element    10. Print the deq     11. Sort  12. Exit
3
0
1. Create deq  2. Is empty    3. Size of the deq     4. Add element to star 5. Add element to the end  6. First element   7.
Last element   8. Delete the first element   9. Delete the last element    10. Print the deq     11. Sort  12. Exit
4
Enter the key: 5
Enter the value: 5.532
1. Create deq  2. Is empty    3. Size of the deq     4. Add element to star 5. Add element to the end  6. First element   7.
Last element   8. Delete the first element   9. Delete the last element    10. Print the deq     11. Sort  12. Exit
4
Enter the key: 7
Enter the value: 7.52352
1. Create deq  2. Is empty    3. Size of the deq     4. Add element to star 5. Add element to the end  6. First element   7.
Last element   8. Delete the first element   9. Delete the last element    10. Print the deq     11. Sort  12. Exit
10
Key    Value
7      7.523520
5      5.532000
1. Create deq  2. Is empty    3. Size of the deq     4. Add element to star 5. Add element to the end  6. First element   7.
Last element   8. Delete the first element   9. Delete the last element    10. Print the deq     11. Sort  12. Exit
4
Enter the key: 1
Enter the value: 1.4121
1. Create deq  2. Is empty    3. Size of the deq     4. Add element to star 5. Add element to the end  6. First element   7.
Last element   8. Delete the first element   9. Delete the last element    10. Print the deq     11. Sort  12. Exit
5
Enter the key: 9
Enter the value: 9.41241
```

1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
10
Key Value
1 1.412100
7 7.523520
5 5.532000
9 9.412410
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
6
Key
1
Value
1.412100
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
7
Key
9
Value
9.412410
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
8
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
9
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
10
Key Value
7 7.523520
5 5.532000
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit

2
Deq is not empty
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
3
2
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
8
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
8
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
2
Deq is empty
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
4
Enter the key: 9
Enter the value: 9.3131
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
5
Enter the key: 11

Enter the value: 11.421
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
5
Enter the key: 10
Enter the value: 10.412
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
5
Enter the key: 1
Enter the value: 1.512
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
5
Enter the key: 4
Enter the value: 4.48493
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
11
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
10

Key	Value
1	1.512000
4	4.484930
9	9.313100
10	10.412000
11	11.421000

1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
9
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
9
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
9
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
9
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
9
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
2
Deq is empty
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
4
Enter the key: 8
Enter the value: 8.423
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
5
Enter the key: 6
Enter the value: 6.42342
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
5
Enter the key: 2
Enter the value: 2.5231

1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
5
Enter the key: 5
Enter the value: 5.52351
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
5
Enter the key: 4
Enter the value: 4.52552
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
5
Enter the key: 7
Enter the value: 7.41442
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
5
Enter the key: 3
Enter the value: 3.5136
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
5
Enter the key: 10
Enter the value: 10.95932
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
5
Enter the key: 1
Enter the value: 1.4198421
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
11
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
10
Key Value
1 1.419842
2 2.523100
3 3.513600
4 4.525520
5 5.523510
6 6.423420
7 7.414420
8 8.423000
10 10.959320
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
8
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
8
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
8
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
8
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
8

1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
8
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
8
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
8
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
8
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
8
Deq is empty
1. Create deq 2. Is empty 3. Size of the deq 4. Add element to star 5. Add element to the end 6. First element 7. Last element 8. Delete the first element 9. Delete the last element 10. Print the deq 11. Sort 12. Exit
12

9. 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

10. **Замечания автора** по существу работы _____
_____
_____

11. **Выводы** Выполнив лабораторную работу, я освоил автоматизацию сборки программ модульной структуры на языке Си с использованием утилиты make, абстрактные типы данных, рекурсию, модульное программирование на языке Си

Недочёты при выполнении задания могут быть устранены следующим образом:
_____
_____
_____

Подпись студента _____