

## Отчет по лабораторной работе № 25 по курсу «Практикум на ЭВМ»

Студент группы Алапанова Эльза Халилевна, № по списку 3

Контакты e-mail : alapanowa02@yandex.ru

Работа выполнена: «21» мая 2021г.

Преподаватель: каф. 806 Найденков Иван  
Евгеньевич

Отчет сдан « » \_\_\_\_\_ 20 \_\_\_\_ г.,  
итоговая оценка \_\_\_\_\_

Подпись преподавателя  
\_\_\_\_\_

**1. Тема:** Абстрактные типы данных. Рекурсия. Модульное программирование на языке Си.

**2. Цель работы:** Реализовать очередь на массиве.

**3. Задание (вариант № 22):** Вставка элемента в очередь. Упорядочить по возрастанию с сохранением порядка. Сортировка простой вставкой.

**4. Оборудование** (студенческое)

Процессор Intel® Core™ i5-10210 @ 1.60 GHz с ОП 8192 Мб, НМД 512 Уб.  
Монитор 1920 x 1080

**5. Программное обеспечение (студенческое):**

Операционная система семейства Ubuntu, наименование Ubuntu 20.04.2 LTS  
версия \_\_\_\_\_

интерпретатор команд \_\_\_\_\_ версия \_\_\_\_\_.

Система программирования \_\_\_\_\_

\_\_\_\_\_ версия \_\_\_\_\_

Редактор текстов \_\_\_\_\_

\_\_\_\_\_ версия \_\_\_\_\_

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

\_\_\_\_\_

Прикладные системы и программы Sublime Text

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

Реализовать очередь на массиве, добавление элемента через переменную qAfter и сортировку вставками.

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

План работы:

- Запуск системы

- Изучение теории

- Написание программы

- Отладка

- Создание протокола

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

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

elza@elza-NBLB-WAX9N:~/26\$ cat main.c

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include "queue.h"
```

```
void init(queue *node) {
```

```
    node->qFirst = 1;
```

```
    node->qAfter = 0;
```

```
    return;
```

```
}
```

```
void insert(queue *node, int key) { // Добавление элемента с клавиатуры key в очередь
```

```
    if(node->qAfter < QMAX-1) {
```

```
        node->qAfter++;
```

```
        node->qu[node->qAfter]=key;
```

```
    }
```

```
    else
```

```
        printf("Очередь полна!\n");
```

```
    return;
```

```
}
```

```
int isempty(queue *node) { // Проверка, пуста ли очередь
```

```
    if(node->qAfter < node->qFirst) return 1;
```

```
    else return 0;
```

```
}
```

```
void print(queue *node) {
```

```
    int h;
```

```
    for(h = node->qFirst; h<= node->qAfter; h++)
```

```
        printf("%d ",node->qu[h]);
```

```
    return;  
}
```

```
int removeL(queue *node) {  
    int x;  
    if(isempty(node)==1) {  
        printf("Очередь пуста!\n");  
        return(0);  
    }  
    x = node->qu[node->qFirst];  
    node->qFirst++;  
    return x;  
}
```

```
void sort(queue *node){  
    int previous;  
    int key;  
    int next;  
    for(next = node->qFirst; next<= node->qAfter; next++) {  
        // i = qu->qFirst; i<= qu->qAfter; i++  
        // i = 0; i < *sizeQueue; i++  
        previous = next - 1;  
        key = node->qu[next];  
        while (node->qu[previous] > key && previous >= 0) {  
            node->qu[previous + 1] = node->qu[previous];  
            previous -= 1;  
        }  
        node->qu[previous + 1] = key;  
    }  
    return;  
}
```

```

int main(){
    queue *q;
    int answer;
    int action;
    q = (queue*)malloc(sizeof(queue));
    while (1)
    {
        printf("Меню:\n");
        printf("1) Создать очередь\n");
        printf("2) Удалить очередь\n");
        printf("3) Сортировка\n");
        printf("4) Выход\n");
        printf("Выберите действие: ");
        scanf("%d", &action);

        switch (action)
        {
            case 1:
            {
                init(q);
                for(int i = 0; i < QMAX; i++) {
                    printf("Введите элемент очереди или число 100 чтобы закончить ввод:
");
                    scanf("%d", &answer);
                    if (answer == 100)
                        break;
                    insert(q, answer);
                }
            }
        }
    }
}

```

```
    print(q);  
    printf("\n");  
    break;  
}
```

case 2:

```
{  
    while(q->qFirst <= q->qAfter) {  
        answer = removeL(q);  
        printf("\nУдален элемент %d\n", answer);  
        print(q);  
    }
```

```
    break;  
}
```

case 3:

```
{  
    sort(q);  
    print(q);  
}
```

case 4: break;

default:

```
{  
    printf("Ошибка. Такого пункта меню не существует\n");  
    break;  
}  
}
```

if (action == 4)

```

        break;
    }
    return 0;
}
elza@elza-NBLB-WAX9N:~/26$ cat queue.h
#ifndef _QUEUE_
#define _QUEUE_

#define QMAX 100

typedef struct{
    int qu[QMAX];
    int qAfter, qFirst;
}queue;

void init(queue *node);
void insert(queue *node, int x);
int isempty(queue *node);
void print(queue *node);
int removeL(queue *node);

#endif
elza@elza-NBLB-WAX9N:~/26$ gcc main.c
elza@elza-NBLB-WAX9N:~/26$ ls
a.out  queue.h          'Алапанова отчет №25.pdf'
main.c 'Алапанова отчет №25.docx' 'Алапанова отчет №26.docx'
elza@elza-NBLB-WAX9N:~/26$ ./a.out

```

Меню:

- 1) Создать очередь
- 2) Удалить очередь
- 3) Сортировка
- 4) Выход

Выберите действие: 1

Введите элемент очереди или число 100 чтобы закончить ввод: 6

Введите элемент очереди или число 100 чтобы закончить ввод: 3

Введите элемент очереди или число 100 чтобы закончить ввод: 5

Введите элемент очереди или число 100 чтобы закончить ввод: 9

Введите элемент очереди или число 100 чтобы закончить ввод: 1

Введите элемент очереди или число 100 чтобы закончить ввод: 100

6 3 5 9 1

Меню:

1) Создать очередь

2) Удалить очередь

3) Сортировка

4) Выход

Выберите действие: 3

1 3 5 6 9 Меню:

1) Создать очередь

2) Удалить очередь

3) Сортировка

4) Выход

Выберите действие: 1

Введите элемент очереди или число 100 чтобы закончить ввод: 4

Введите элемент очереди или число 100 чтобы закончить ввод: 1

Введите элемент очереди или число 100 чтобы закончить ввод: 876

Введите элемент очереди или число 100 чтобы закончить ввод: 4

Введите элемент очереди или число 100 чтобы закончить ввод: 35

Введите элемент очереди или число 100 чтобы закончить ввод: 100

4 1 876 4 35

Меню:

1) Создать очередь

2) Удалить очередь

3) Сортировка

4) Выход



Выберите действие: 3

1 4 4 35 876 Меню:

- 1) Создать очередь
- 2) Удалить очередь
- 3) Сортировка
- 4) Выход

Выберите действие: 1

Введите элемент очереди или число 100 чтобы закончить ввод: 9

Введите элемент очереди или число 100 чтобы закончить ввод: 24

Введите элемент очереди или число 100 чтобы закончить ввод: 75

Введите элемент очереди или число 100 чтобы закончить ввод: 38

Введите элемент очереди или число 100 чтобы закончить ввод: 2

Введите элемент очереди или число 100 чтобы закончить ввод: 100

9 24 75 38 2

Меню:

- 1) Создать очередь
- 2) Удалить очередь
- 3) Сортировка
- 4) Выход

Выберите действие: 3

2 9 24 38 75 Меню:

- 1) Создать очередь
- 2) Удалить очередь
- 3) Сортировка
- 4) Выход

Выберите действие: 4

elza@elza-NBLB-WAX9N:~/26\$ Реализовать очередь на массиве, добавление элемента через переменную qAfter и сортировку вставками.

Реализовать: команда не найдена

elza@elza-NBLB-WAX9N:~/26\$

**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.

№	Ла б. ил и до м.	Да та	Вре мя	Событие	Действие по исправлению	Примечание
1						
2						

**10. Замечания автора** по существу работы : замечаний нет.

**11. Выводы :** узнала о этапах сборки программных модулей,смогла автоматизировать сборку с помощью take. Реализовала модуль дека и выполнила над ним сортировку. Недочёты при выполнении задания могут быть устранены следующим образом:

\_\_\_\_\_

\_\_\_\_\_Подпись  
студента \_\_\_\_\_