**Отчет по лабораторной работе №25-26** по курсу\_\_\_\_\_\_1\_\_\_\_\_\_\_

Студент группы: **М8О-101Б-22**, **Соколова Виктория Дмитриевна**, № по списку: **20**, Контакты**,** [viktoriyasokolova04@gmail.com](mailto:viktoriyasokolova04@gmail.com) telegram **@Angrytea\_01** Работа выполнена: « » апрель 2023г.

Преподаватель: **каф. 806 Крылов Сергей Сергеевич**, Входной контроль знаний с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Отчет сдан « » 2023 г., итоговая оценка \_\_\_\_\_

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

1. **Тема:** Алгоритмы и структуры данных
2. **Цель работы:** Составить и отладить модуль определений и модуль реализации по заданной схеме модуля определений для абстрактного типа данных.
3. **Задание** (*вариант № 20* )**:** Поиск в очереди двух элементов. Идущих подряд, первый из которых больше второго. Если такие элементы найдены, их перестановка. Сортировка методом пузырька.
4. **Оборудование** (лабораторное):

ЭВМ , процессор , имя узла сети с ОП Мб, НМД Мб. Терминал адрес . Принтер Другие устройства

**Оборудование ПЭВМ студента, если использовалось:**

Процессор **Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz 2.50 GHz** с ОП **16384** Мб, ТТН **524 288** Мб. Встроенный монитор 15.6 дюймов диагональ, разрешение **1920 \* 1080**.

Другие устройства

1. **Программное обеспечение ЭВМ студента, если использовалось:**

Операционная система семейства **Linux** , наименование **Linux Ubuntu** версия 22.04.01 интерпретатор команд **bash** версия **5.1.16** Система программирования версия Редактор текстов **nano** версия **6.2** 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 постусловиям.

1. Файл 26.c содержит main()

2. functions.c – реализация

3. functions.h - интерфейс

4. Makefile.mak – сценарий компиляции и редактирования связей

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

26.с

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include "functions.h"

int main(){

    char tmp;

    int a, ch = 0, i, l = 0, g = 0;

    char m[100];

    que \*Q = malloc(sizeof(que));

    init(Q);

    while (1){

        printf("\n");

        printf("1 - dobavit znach v ochered\n");

        printf("2 - sortirovka 1 elementa\n");

        printf("3 - print\n");

        printf("4 - exit\n");

        scanf("%s", &m);

        printf("\n");

        if (strcmp(m, "1") == 0){

            printf("vvedi: ");

            scanf("%d", &a);

            push(Q, a);

            l++;

        }

        if (strcmp(m, "2") == 0){

            if (g == l){

                printf("otsortirovana blin\n");

                continue;;

            }else if(g != l){

                g = 0;

            }

            for (int i = 0; i < l; i++){

                push(Q, del(Q));

                ch++;

                g++;

                if (Q->data[Q->tail] > Q->data[Q->head]){

                    if (ch == l){

                        printf("otsortirovana blin\n");

                        break;

                    }

                    tmp = Q->data[Q->tail];

                    Q->data[Q->tail] = Q->data[Q->head];

                    Q->data[Q->head] = tmp;

                        for(int z = 0; z < l-ch; z++){

                            push(Q, del(Q));

                    }

                    ch = 0;

                    print(Q);

                    printf("\n");

                    break;

                }else if (Q->data[Q->tail] <= Q->data[Q->head]){

                    continue;

                }

            }

        }

        if (strcmp(m, "3") == 0){

            print(Q);

        }

        if (strcmp(m, "4") == 0){

            break;

        }

    }

}

functions.h

#ifndef FUNC\_H

#define FUNC\_H

typedef struct que{

    int data[100];

    int size, head, tail;

}que;

void init(que \*Q);

void push(que \*Q, int x);

int del(que \*Q);

void print(que \*Q);

int isempty(que \*Q);

#endif

functions.c

#include "functions.h"

#include <stdio.h>

void init(que \*Q){

    Q->head = 1;

    Q->tail = 0;

    Q->size = 2;

}

int isempty(que \*Q) {

    if(Q->tail < Q->head) {

       return 1;

    }

    return 0;

}

void push(que \*Q, int x){

    if (Q->size == 100){

        printf("polnaya ochered\n");

        return;

    }

    Q->tail++;

    if (Q->tail >= 100){

        Q->tail = 0;

    }

    Q->data[Q->tail] = x;

    Q->size++;

}

int del(que \*Q){

    int temp;

    if (Q->size == 0){

        printf("pysto\n");

        return 23456;

    }

    temp = Q->data[Q->head];

    Q->head++;

    if (Q->head >= 100){

        Q->head = 0;

    }

    Q->size--;

    return temp;

}

void print(que \*Q) {

  int h;

  if(isempty(Q)==1) {

    printf("Очередь пуста!\n");

    return;

  }

  for(h = Q->head; h<= Q->tail; h++)

    printf("%d ",Q->data[h]);

  return;

}

Makefile.mak

all: 26

26: 26.o functions.o

    gcc 26.o functions.o -o 26

26.o: 26.c

    gcc -c 26.c

functions.o: functions.c

    gcc -c functions.c

clean:

    rm -rf \*.o 26

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

1. **Распечатка протокола:**

PS C:\code\test\25-26> make -f Makefile.mak

gcc -c 26.c

gcc -c functions.c

gcc 26.o functions.o -o 26

PS C:\code\test\25-26> ./26

1 - dobavit znach v ochered

2 - sortirovka 1 elementa

3 - print

4 - exit

1

vvedi: 3

1 - dobavit znach v ochered

2 - sortirovka 1 elementa

3 - print

4 - exit

1

vvedi: 2

1 - dobavit znach v ochered

2 - sortirovka 1 elementa

3 - print

4 - exit

1

vvedi: 5

1 - dobavit znach v ochered

2 - sortirovka 1 elementa

3 - print

4 - exit

1

vvedi: 4

1 - dobavit znach v ochered

2 - sortirovka 1 elementa

3 - print

4 - exit

3

3 2 5 4

1 - dobavit znach v ochered

2 - sortirovka 1 elementa

3 - print

4 - exit

2

2 3 5 4

1 - dobavit znach v ochered

2 - sortirovka 1 elementa

3 - print

4 - exit

2

2 3 4 5

1 - dobavit znach v ochered

2 - sortirovka 1 elementa

3 - print

4 - exit

2

otsortirovana blin

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
2. **Выводы**: Я составила и отладила модуль определений и модуль реализации по заданной схеме модуля определений для абстрактного типа данных.

Недочёты при выполнении задания могут быть устранены следующим образом:

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