МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

ОТЧЕТ

по лабораторной работе №3

по дисциплине «Алгоритмы и структуры данных»

Тема: Линейные структуры данных: стек, очередь, дек

Студент гр. 7383 ___ Сычевский Р.А.

Преподаватель ___ Размочаева Н.В.

Санкт-Петербург

2018

Содержание

1. ЦЕЛЬ РАБОТЫ

Цель работы: Познакомиться с часто используемыми на практике линейными структурами данных, обеспечивающими доступ к элементам последовательности только через её начало и конец, и способами реализации этих структур, освоить на практике использование стека, очереди и дека для решения задач.

Формулировка задания (Вариант 2): Содержимое заданного текстового файла F, разделенного на строки, переписать в текстовый файл G, перенося при этом в конец каждой строки все входящие в неё цифры (с сохранением исходного взаимного порядка как среди цифр, так и среди остальных литер строки).

2. РЕАЛИЗАЦИЯ ЗАДАЧИ

В данной работе используется главная функция (main()) и класс Queue с функциями add_elem(), rem_elem() и count()).

Сначала функция main() спрашивает у пользователя имя файла, в котором записаны необходимые строки (input файл). После этого программа считывает строчки из файла и находя цифры записывает их в очередь, построенную на базе массива, и перезаписывает строку. Доходя до конца строки, программа берет цифры из очереди и приписывает в конец строки. После этого программа записывает строку в output файл.

3. ТЕСТИРОВАНИЕ

3.1 ПРОЦЕСС ТЕСТИРОВАНИЯ

Программа собрана в операционной системе Ubuntu 18.04.1 LTS bionic компилятором g++ (Ubuntu 7.3.0-16ubuntu3) 7.3.0. В других ОС и компиляторах тестирование не проводилось.

3.2 РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

Тестовые случаи представлены в Приложении А.

Результаты тестирования показали, что программа работает верно, значит поставленная задача выполнена.

4. ВЫВОД

В ходе выполнения данной работы были изучены линейные структуры данных, такие как стек, очередь и дек. Была написана программа, в которой реализована очередь на основе массива.

ПРИЛОЖЕНИЕ А: ТЕСТОВЫЕ СЛУЧАИ

№	Ввод	Вывод
1	sdfghj	sdfghj
2	erthjnbv	erthjnbv
3	erty2	erty2
4	234rfvtb	rfvtb234
5	w2e3r4t5	wert2345
6	345tf34	tf34534
7	12345678	12345678

ПРИЛОЖЕНИЕ Б: КОД ПРОГРАММЫ

```
#include <iostream>
#include <cctype>
#include <fstream>
#include <sstream>
#include <cstring>
using namespace std;

class Queue{
    private:
    int first;
    int last;
    int *arr;
    int max;
    void resize(){
        max += 5;
        int *tmp = new int[max];
        for(int i = first; i <=last; i++)
            tmp[i] = arr[i];
        delete [] arr;
        arr = tmp;
    }
    public:
    Queue(){
        max = 10;
        arr = new int[max];
        first = 0;
        last = -1;
    }
    void add_elem(int a){
        if(last == max-1)
            resize();
        last++;
        arr[last] = a;
    }
    int rem_elem(){
        if(count()){
            first++;
            return arr[first-1];
        } else
            cout << "массив пуст!" << endl;
    return 0;
    }
    int count(){
        return last-first+1;
    }
    ~Queue(){
        delete [] arr;
    }
};

int main(){
    string tmp;
    string file_name;
```

```
    cout << "Введите имя файла" << endl;
    cin >> file_name;
    ifstream f1;
    f1.open(file_name.c_str());
    if (!f1){
        cout << "Файл не открыт!" << endl;
        return 0;
    }
    ofstream f2;
    f2.open("output.txt");
    getline(f1, tmp);
    while(!f1.eof()){
        Queue arr1;
        for(int i = 0; i < tmp.length(); i++){
            if((tmp[i] > 47) && (tmp[i] < 58)){
                arr1.add_elem(tmp[i]);
                for(int j = i; j < tmp.length()-1-arr1.count(); j++)
                    tmp[j] = tmp[j+1];
                    if(i == tmp.length()-1-arr1.count())
                        break;
                    i--;
            }
        }
        while(arr1.count()){
            tmp[tmp.length()-1-arr1.count()] = arr1.rem_elem();
        }
        f2 << tmp << endl;
        getline(f1, tmp);
    }
    f1.close();
    f2.close();
    return 0;
}
```