

Министерство науки и высшего образования Российской
Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)
Кафедра автоматизированных систем управления (АСУ)

ТЕМА ЛАБОРАТОРНОЙ РАБОТЫ – СТЕКИ И ОЧЕРЕДИ

**Отчет по лабораторной работе №1
По дисциплине
«Структуры и алгоритмы обработки данных в ЭВМ»**

Студент гр. 431-Х

_____ А.В. Гурулёв
« ____ » _____ 2022 г.

Проверил: профессор кафедры АСУ, д.т.н.

_____ А.Н. Горитов
« ____ » _____ 2022 г.

Томск 2022

1 Задание на лабораторную работу

Разработать программу формирования стека, куда помещаются целые положительные числа, которые вводятся из файла. Процесс ввода должен прекращаться как только появляется отрицательное число. После этого программа должна вывести на экран монитора содержимое стека, при этом порядок выводимых чисел должен быть обратным по сравнению с последовательностью их ввода.

Для реализации АДТ Стек использовать массив.

2 Алгоритм решения задачи

1. Открываем файл;
2. Пока не встретится отрицательное число, заносим числа в стек;
3. Забираем из стека элементы, пока он не станет пустым.

3 Листинг программы

Для main.cpp:

```
#include <iostream>
#include <locale.h>
#include <fstream>

#include "stackLogic.cpp"

using namespace std;
using namespace myTaskLogic;

int main()
{
    setlocale(LC_ALL, "rus");

    MyStack stack;

    ifstream f("File.txt");

    if (f.is_open())
    {
        //Заполним стек, а затем выведем его элементы
        int num = 0;
        while (f >> num)
        {
            if (num >= 0)
            {
                stack.Push(num);
            }
            else
            {
                break;
            }
        }

        while (!stack.IsEmpty())
        {
```

```

        int out = stack.Pop();
        cout << out << " ";
    }
}
else
{
    cout << "error" << endl;
}

system("pause");

return 0;
}

```

Для stackLogic.cpp:

```

#include <iostream>

using namespace std;

namespace myTaskLogic
{
    class MyQueue
    {
    public:
        explicit MyQueue()
        {
            size = 100;
            start = 0;
            end = -1;
            queue = (int*)malloc(sizeof(int) * size);
        }
        ~MyQueue()
        {
            delete[] queue;
            start = 0;
            end = 0;
        }

        //Поместить элемент в очередь
        void Push(int num)
        {
            if (end == (size - 1))
            {
                cout << "Очередь полна" << endl;
                exit(EXIT_FAILURE);
            }
            else
            {
                queue[++end] = num;
            }
        }

        //Изъять элемент из очереди
        int Pop()
        {
            if (start > end)
            {
                cout << "Очередь пуста" << endl;
                exit(EXIT_FAILURE);
            }
            else
            {
                return queue[start++];
            }
        }
    };
}

```

```

    }
}

//Получение первого
int Start()const
{
    return queue[start];
}

//Является ли очередь пустой
bool IsEmpty()const
{
    if (start > end)
    {
        return true;
    }
    else
    {
        return false;
    }
}

private:
    short int size;
    short int start;
    short int end;
    int* queue;
};
}

```

4 Пример решения

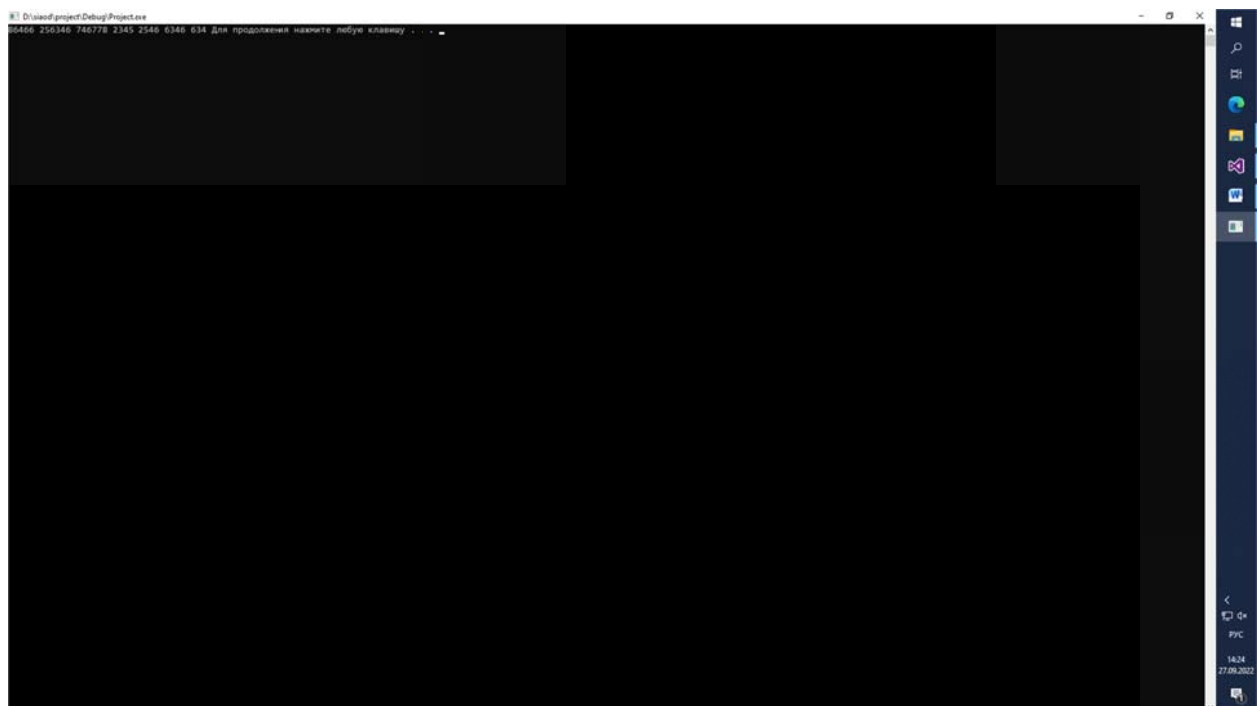


Рисунок 4.1 - Результат выполнения программы



Рисунок 4.2 - Входные данные

5 Вывод

Я изучил возможности стека, а так же научился его использовать.