МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

«ТОМСКИЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОННИКИ»(ТУСУР)

Кафедра автоматизированных систем управления (АСУ)

Отчёт по лабораторной работе №1

«Стеки и очереди»

по дисциплине «Структуры и алгоритмы обработки данных в ЭВМ»

Профессор кафедры (АСУ)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Горитов А.Н.

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 г.

Студент гр. 438-3

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Канаев О.А.

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_2019г

2019

**Цель работы:**Научиться работать с АТД Стек, используя массив.

**Задание:**Используя стек, решить следующую задачу. Напечатать каждое слово текстового файла T в обратном порядке. Разделителями слов могут быть пробелы, запятые и точки.

**Алгоритм:**

Реализовываем стек по типу одномерного статического массива типа string. Следущее реализовываем стандартные функции работы со стеком: Push, Pop, Top, IsEmpty, StackOverflow.

Затем реализовываем функцию по карточке. Программа по очереди берет каждое слово из стека, записывает их в обратном порядке.

**Решение:**#include <iostream>

#include <fstream>

#include <string>

#include <cstring>

#include <cstdio>

#define MAX\_Elem 5

using namespace std;

bool IsEmpty(int &count);

bool StackOverflow(int &count);

void Push(string &word, int &count, int &start, string \*stack);

string Pop(int &start, int &count, string \*stack);

void Top(int &start, int&count, string\* stack);

void Reverse(int &start, int&count, string\* stack);

int main() {

string stack[MAX\_Elem];

int count = -1, start=MAX\_Elem;

Reverse(start, count, stack);

string a= "hello", c="programm";

Push(a, count, start, stack);

Push(c, count, start, stack);

//cout << Pop(start, count, stack) << endl;

Top(start, count, stack);

system("pause");

return 0;

}

bool IsEmpty(int &count) {

if ((count > -1) && (count<=MAX\_Elem)) return false;

else return true;

}

bool StackOverflow(int &count) {

if (count >= MAX\_Elem)

{

cout << "Stack overflow!" << endl;

return true;

}

else return false;

}

void Push(string &word, int &count, int &start, string \*stack) {

if (count==-1)count++;

if (StackOverflow(count)==true) return;

stack[MAX\_Elem-count-1] = word;

start--;

if (count > -1) count++;

return;

}

string Pop(int &start, int &count, string \*stack) {

string word;

if (IsEmpty(count) == true) return word;

word=stack[start];

stack[start] = "\0";

start++;

count--;

return word;

}

void Top(int &start, int&count, string\* stack) {

cout << stack[start] << endl;

return;

}

void Reverse(int &start, int&count, string\* stack)

{

int size=0, size\_const=size;

char word[50], sbl;

FILE \*file = fopen("T.txt", "r+");

if (!file) return;

if (count == -1) count++;

while (!feof(file))

{

sbl = fgetc(file);

if ((sbl != ' ') && (sbl != '.') && (sbl != ',') && (!feof(file)))

{

fseek(file, 1 / 2, SEEK\_CUR);

}

else if (!feof(file)) {

fseek(file, -1 / 2, SEEK\_CUR);

size = ftell(file)-size\_const;

size\_const += size;

fseek(file, -size, SEEK\_CUR);

fgets(word, size, file);

rewind(file);

fseek(file, size\_const, SEEK\_CUR);

stack[MAX\_Elem - count - 1] = string(word);

start--;

count++;

}

else {

fseek(file, -1 / 2, SEEK\_CUR);

size = ftell(file) - size\_const;

size\_const += size;

fseek(file, -size, SEEK\_CUR);

fgets(word, size+1, file);

rewind(file);

fseek(file, size\_const, SEEK\_CUR);

stack[MAX\_Elem - count - 1] = string(word);

start--;

count++;

break;

}

}

for (int i = start; i < MAX\_Elem; i++)

{

for (int j = stack[i].length() - 1; j >= 0; j--)

{

cout << stack[i].at(j);

}

stack[i] = "\0";

cout << " ";

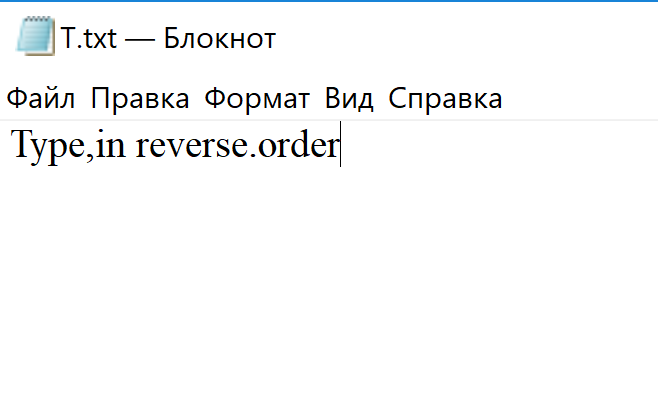
}

cout << "\n";

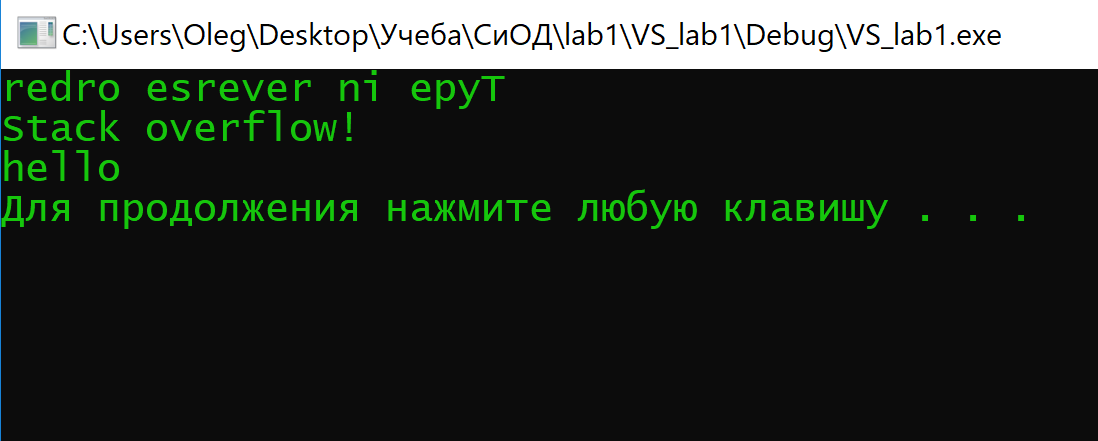
}

**Результат работы:**

Исходный файл T.txt:



Работа программы:



**Вывод:**

Я научился работать с АТД Стек, изучил и реализовал функции для работы со стеком используя массив типа string и стандартные функции работы с ним.