

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

ОТЧЕТ
по лабораторной работе №3
по дисциплине «Алгоритмы и структуры данных»
Тема: Рекурсия

Студент гр. 8304

Мухин А. М.

Преподаватель

Фирсов М. А.

Санкт-Петербург

2019

Цель работы.

Ознакомиться с основными понятиями и приёмами рекурсивного программирования, получить навыки программирования рекурсивных процедур и функций на языке программирования C++.

Задание.

Содержимое заданного текстового файла F , разделенного на строки, переписать в текстовый файл G , выписывая литеры каждой строки в обратном порядке.

Вариант 4-в.

Выполнение работы.

В заголовочном файле объявлен класс `Stack`. В частных полях которого есть указатель типа `type` и порядковый номер текущего элемента. Методы описаны в открытой части класса. Конструктор выделяет память в указатель типа `type`. Деструктор удаляет содержимое этого указателя и зануляет его ссылку.

Метод `void push(type)` записывает в массив элемент типа `type`.

Метод `type pop()` возвращает верхний элемент стэка.

Метод `bool isEmpty()` возвращает истину, если стэк пуст.

Функция `void Reverse(Stack<type>&, std::ifstream&, std::ofstream&, char&)` производит чтение из одного файла и запись в другой согласно заданию. Если текущий символ потока это не перенос строки, тогда мы его засовываем в стэк и снова вызываем эту функцию. Если это символ переноса строки, тогда мы достаём последний элемент из стэка и записываем его в поток вывода.

Функция `int main(int, char*)` выполняет считывание файла первый раз, для того, чтобы выяснить количество строк и длину максимальной строки, для дальнейшей инициализации стэка. После этого считывания, поток возвращается в начало файла и в цикле `for`, количество строк в файле раз объявляется переменная типа `Stack` и вызывается функция `void`

Reverse(Stack<type>&, std::ifstream&, std::ofstream&, char&). В конце происходит закрытие потока ввода и вывода. Стоит отметить, что поток вывода открывался с флагом std::ios::app, благодаря которому, запись в файл происходила в его конец.

Разработанный программный код см. в приложении А.

Тестирование.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные
1.	kjhgvbjhlb nljhnk nluikhn	blhjbvghjk knhjln nhkiuln
2.	sfgsdfgsdf gs dfgsdsgsdfgs fgsdfg sdfgs sdf s dfgsdfgsdfg	fdsgfdsgfs sg sgfdsgsdsgfd gfdsgf sgfds fds s gfdsgfdsgfd
3.	sfgsdfgsdf gssdfgsdfgsdfg dfgsdsgsdfgs fgsdfg fgsdfgsdfgsdfg sdfgssdfgsdfgsdfg sdf sgfsfgsfgsdfgsdf dfgsdfgsdfg	fdsgfdsgfs gfdsgfdsgfdssg sgfdsgsdsgfd gfdsgf gfdsgfdsgfdsgf gfdsgfdsgfdssgfd fds fdsgfdsgfsgfsfgs gfdsgfdsgfd

4.		
5.	a b c	a b c
6.	4a1234 bfadsfasef cflksufdhgnldf sd;ifuhglsdf adflghbsdljfg sdflgisbdfnl sdfmglisdnfgls fngouayhbdfgklj adfmkgisdfhunlgs alfnidugnaslifd afmgiluafhnglaidf afdnlgiauabnldfg dflnugnself	4321a4 fesafsda fb fdlnghdfusk lfc fdslghufi;ds gfjlds b h jglfda lnfdbsigl fds slgfndsilgm fds jlk g fdbhyauognf sglnuhfdsiglm fda dfilsangudin fla fdialgnhfauligmfa gfdlnbauiglndfa flosngunlfd

Выводы.

Ознакомились с основными понятиями и приёмами рекурсивного программирования, получили навыки программирования рекурсивных процедур и функций на языке программирования C++ и создания стэка на базе вектора. Научились работать с потоками ввода и вывода и создавать шаблоны.

ПРИЛОЖЕНИЕ А ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab3.cpp

```
#include "lab3.h"
#include <iostream>
```

```

#include <fstream>

template <class type>
Stack<type>::Stack(size_t max_size) {
    st = new type[max_size + 2];
    len = 0;
}

template <class type>
Stack<type>::~~Stack() {
    delete [] st;
    st = nullptr;
}

template <class type>
void Stack<type>::push(type elem) {
    st[len] = elem;
    len++;
}

template <class type>
type Stack<type>::pop() {
    return st[--len];
}

template <class type>
bool Stack<type>::isEmpty() {
    return len == 0;
}

template <class type>
void Reverse(Stack<type>& stack, std::ifstream& in,
std::ofstream& out, char& s) {
    if (s != EOF) {
        if (s != '\n') {
            stack.push(s);
            in.get(s);
            if (!in.eof())
                Reverse(stack, in, out, s);
        }
        if (!stack.isEmpty()) {
            out.put(stack.pop());
            return;
        }
    }
}

int main(int argc, char* argv[]) {
    std::ifstream in(argv[1]);
    std::ofstream out(argv[2], std::ios::app);
}

```

```

int maxLen = 0;
int countString = 0;
std::string currentLine;
while (getline(in, currentLine)) {
    countString++;
    if (currentLine.length() > maxLen)
        maxLen = currentLine.length();
}
in.clear();
in.seekg(0, std::ios::beg);
char s;
for (int i = 0; i < countString; i++) {
    Stack<char> stack(maxLen);
    in.get(s);
    if (s != EOF) {
        Reverse(stack, in, out, s);
        out.put('\n');
    }
}
in.close();
out.close();
return 0;
}

```

Название файла: lab3.h

```

#ifndef LAB3
#define LAB3
#include <cstdlib>
template <class type>
class Stack{
    private:
        type* st;
        size_t len;
    public:
        Stack(size_t max_size);
        ~Stack();
        void push(type elem);
        type pop();
        bool isEmpty();
};
#endif

```