

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

ОТЧЕТ
по лабораторной работе №1
по дисциплине «Алгоритмы и структуры данных»
Тема: Рекурсия

Студент гр. 9382

Рыжих Р.В.

Преподаватель

Фирсов М.А.

Санкт-Петербург

2020

Цель работы.

Изучение и приобретение практических навыков работы с рекурсивными функциями.

Основные теоретические положения.

Написание кода производилось на базе системы Windows 10. Код был написан на языке C++ с использованием исключительно стандартных библиотек. Работа с кодом происходила в IDE Microsoft Visual Studio.

Задание.

17. Функция Φ преобразования текста определяется следующим образом (аргумент функции – это текст, т. е. последовательность символов):

$$\Phi(\alpha) = \begin{cases} \Phi(\gamma)\beta, & \text{если } \alpha = \beta/\gamma \text{ и текст } \beta \text{ не содержит вхождений символа «/»}, \\ \alpha, & \text{если в } \alpha \text{ нет вхождений символа «/»}. \end{cases}$$

Например: $\Phi(\text{«ла/ска»}) = \text{«скала»}$, $\Phi(\text{«б/ру/с»}) = \text{«сруб»}$, $\Phi(\text{«ца/ри/ца»}) = \text{«царица»}$, $\Phi(\text{«ум/ри/ва/к/а»}) = \text{«аквариум»}$. Реализовать функцию Φ рекурсивно.

Выполнение работы.

1) В функции `main` создаются две строки `a` и `answ`, для записи и для вывода соответственно. Затем открываются файлы для записи и для вывода, и в строку `a` считывается строка из входного файла. После, строка подаётся в рекурсивную функцию `func` вместе с переменной `x`, отвечающей за отступы, и указателем на строку вывода `answ`.

2) В рекурсивной функции `func` исходная строка разбивается на две строки: одна до символа `“/”`, вторая – после него; если в исходной строке символа `“/”` не было, то возвращается сама строка. Затем, данная функция применяется ко второй строке, а в строку вывода возвращается строка до символа `“/”`.

3) Результат работы программы выводится в файл вывода. Вся отладочная информация выводится в консоль. В ней показано, с какой строкой в данный момент работает программа и какую строку она возвращает. Для более понятного вида, показана глубина рекурсии с помощью функции `printer`, которая принимает в себя переменную `x`, увеличивающуюся с увеличением глубины рекурсии, и выводит необходимое количество отступов, соответствующих текущей глубине.

Тестирование.

Результаты тестирования представлены в таблице 1.

Таблица 1 — Результаты тестирования

No	Входные данные	Выходные данные	Комментарии
1	!/rld/Wo/ /lo/Hel	В консоль: Now working with string: !/rld/Wo/ /lo/Hel Now working with string: rld/Wo/ /lo/Hel Now working with string: Wo/ /lo/Hel Now working with string: /lo/Hel Now working with string: lo/Hel Now working with string: Hel Returning string: Hel Returning string: lo Returning string: Returning string: Wo Returning string: rld Returning string: ! В файл: Answer is: Hello World!	Пробелы также считаются символом строки
2	ka/ba/bu	В консоль: Now working with string: ka/ba/bu Now working with string: ba/bu Now working with string: bu Returning string: bu Returning string: ba Returning string: ka В файл: Answer is: bubaka	
3	Hello	В консоль: Now working with string: Hello Returning string: Hello В файл: Answer is: Hello	В строке без символа “/” возвращается сама строка

Вывод.

Были изучена и опробована работа с рекурсивными функциями на языке C++. Была создана программа, использующая рекурсивные функции.

ПРИЛОЖЕНИЕ С КОДОМ

main.cpp :

```
#include <iostream>
#include <string>
#include <fstream>

using namespace std;

void printer(int x)
{
    for (int i = 0; i < x; i++)
        cout << "    ";
    return;
}

void func(string a, int x, string* answ)
{
    printer(x);
    cout << "Now working with string: " << a << endl;
    if (a.empty())
    {
        *answ = a;
        printer(x);
        cout << "Returning string: " << a << endl;
        return;
    }
    if (a.find('/') == string::npos)
    {
        *answ = a;
        printer(x);
        cout << "Returning string: " << a << endl;
        return;
    }
    string b;
    b.append(a, a.find('/')+1);
    a.erase(a.find('/'));
    func(b, x+1, answ);
    *answ += a;
    printer(x);
    cout << "Returning string: " << a << endl;
    return;
}

int main()
{
    string a, answ;
    int x = 0;
    ifstream fin("input.txt");
    ofstream fout("output.txt");
    getline(fin, a);
    func(a, x, &answ);
    fout << "Answer is: " << answ << endl;
    fin.close();
    fout.close();
    return 0;
}
```