

**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. 9304

Попов Д.С.

Преподаватель

Фиалковский М.С.

Санкт-Петербург

2020

### Цель работы.

Изучить понятие рекурсии и методы её применения.

Научиться грамотно реализовывать программы с использованием рекурсии в языке C++.

### Задание.

18. Пусть определена функция  $\Phi$  преобразования целочисленного вектора  $\alpha$ :

$$\Phi(\alpha) = \begin{cases} \alpha, & \text{если } \|\alpha\| = 1, \\ ab, & \text{если } \|\alpha\| = 2, \alpha = ab \text{ и } a \leq b, \\ ba, & \text{если } \|\alpha\| = 2, \alpha = ab \text{ и } b < a, \\ \Phi(\beta)\Phi(\gamma), & \text{если } \|\alpha\| > 2, \alpha = \beta\gamma, \text{ где } \|\beta\| = \|\gamma\| \text{ или } \|\beta\| = \|\gamma\| + 1. \end{cases}$$

Например:  $\Phi(1,2,3,4,5) = 1,2,3,4,5$ ;  $\Phi(4,3,2,1) = 3,4,1,2$ ;  $\Phi(4,3,2) = 3,4,2$ . Отметим, что функция  $\Phi$  преобразует вектор, не меняя его длину. Реализовать функцию  $\Phi$  рекурсивно.

### Выполнение работы.

Программа принимает в качестве аргументов два параметра: текстовый файл на чтение и текстовый файл на запись. Из первого считывает набор целочисленных значений и инициализирует ими массив. Знаки пунктуация, буквы, спецсимволы игнорируются. Затем она начинает попарно сравнивать элементы массива. Обход массива происходит с помощью рекурсивной лямбды. В лямбда-захваты передаются ссылки на массив наших значений (*myVector*), массив строк для ведения логов (*logs*) и значение для определения глубины рекурсии (*deep*). В качестве аргументов подается позиция в массиве (*from*), размер необработанной части массива (*size*) и указатель на нашу лямбду (*RF*). Выход из рекурсии происходит при отработке всего массива. Каждая отработка рекурсии сопровождается записью в логи, которые хранятся в текстовом файле указанном в качестве второго аргумента программы.

Разработанный программный код см. в приложении А.

### **Тестирование.**

Для проведения тестирования был написан bash-скрипт ./script .Скрипт запускает программу где в качестве входных аргументов служат заранее подготовленные файлы, расположенные в папке ./Tests

Результаты тестирования см. в приложении Б.

### **Выводы.**

Было изучено понятие рекурсии, а также методы её применения.

Была реализована программа на языке C++ с использованием рекурсивного алгоритма.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.cpp

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

int main(int argc, char* argv[]){

    setlocale(LC_ALL, "ru");
    if(argc != 3){
        std::cout << "Неверное кол-во аргументов!" << std::endl;
        return 1;
    }

    std::ifstream test(argv[1]);
    std::ofstream result(argv[2]);

    if(!test){
        std::cout << "Файл " << argv[1] << " не может быть открыт
на чтение!" << std::endl;
        return 1;
    }

    std::vector<int> myVector {};
    std::vector<std::string> logs;

    std::string strInput;
    getline(test, strInput);
    strInput.c_str();
    bool flag = 0;
    for(int i = 0; strInput[i] != '\0'; i++){
        if(isdigit(strInput[i])){
            if(flag){
                continue;
            }else{
                myVector.emplace_back(atoi(&strInput[i]));
                flag = 1;
            }
        }else{
            flag = 0;
        }
    }

    int deep = 0;

    for(int i = 0; i < myVector.size(); i++){
        logs.emplace_back(std::to_string(myVector[i]) + ' ');
    }
    logs.emplace_back("<- Начальный вектор\n");
```

```

        auto RF = [&myVector, &logs, &deep](size_t from, size_t size,
auto&& RF){
    deep++;
    if(size == 0){
        deep--;
        return;
    }
    if(size > 2){
        RF(from + 2, size - 2, RF);
    }
    if(myVector[from] > myVector[from + 1]){
        std::swap(myVector[from], myVector[from + 1]);
    }
    std::string new_data{};
    for(int i = 0; i < myVector.size(); i++){
        new_data += std::to_string(myVector[i]);
    }
    logs.emplace_back("глубина[" + std::to_string(deep) +
"] = " + std::move(new_data) + ' ' + '\n');
    deep--;
};

    if(myVector.size() % 2){
        RF(0, myVector.size() - 1, RF);
    }else{
        RF(0, myVector.size(), RF);
    }

    for(const auto &str : logs){
        result << str;
    }

    test.close();
    result.close();

    return 0;
}

```

Название файла: script

```

#!/bin/bash
arg1=$(cat Tests/test1.txt)
echo "Test 1:"
echo "argument = $arg1"
./lab1 ./Tests/test1.txt result1.txt
cat result1.txt

echo -e "\nTest 2:"
arg2=$(cat Tests/test2.txt)
echo "argument = $arg2"
./lab1 ./Tests/test2.txt result2.txt
cat result2.txt

echo -e "\nTest 3:"
arg3=$(cat Tests/test3.txt)
echo "argument = $arg3"
./lab1 ./Tests/test3.txt result3.txt
cat result3.txt

```

## ПРИЛОЖЕНИЕ Б

### ТЕСТИРОВАНИЕ ПРОГРАММЫ

Результаты тестирования представлены в табл.1

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	9 2 5 1	9 2 5 1 <- Начальный вектор глубина[2] = 9215 глубина[1] = 2915	9 2 и 5 1 поменялись местами
2.	3,14 2.73	3 14 2 73 <- Начальный вектор глубина[2] = 314273 глубина[1] = 314273	Ничего не изменилось, числа в правильном порядке
3.	9 clava ukraine 3 7 5	9 3 7 5 <- Начальный вектор глубина[2] = 9357 глубина[1] = 3957	Игнор букв