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

ОТЧЕТ

по лабораторной работе №1

по дисциплине «Алгоритмы и структуры данных»

Тема: Рекурсия

Студент гр. 7383 ___ Сычевский Р.А.

Преподаватель ___ Размочаева Н.В.

Санкт-Петербург

2018

Содержание

1. ЦЕЛЬ РАБОТЫ

Цель работы: познакомиться с основными понятиями и приемами рекурсивного программирования, получить навыки программирования рекурсивных процедур и функций на языке программирования C++.

Формулировка задачи: реализовать рекурсивно функцию Ф преобразования целочисленного вектора α такую, что:

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

2. РЕАЛИЗАЦИЯ ЗАДАЧИ

В данной работе используется главная функция (main()) и дополнительные функции (pr_menu(), sort_v(), work_with_console(), work_with_file()).

Сначала функция main() вызывает функцию pr_menu(), которая используется для вывода меню. Пользователь выбирает какой-нибудь пункт в меню и вводит число, которое передается в switch(). В зависимости от выбранного пункта выбирается необходимая опция:

1 – ввод массива с клавиатуры;

2 – считывание массива из файла;

0 – выход из программы;

Если пользователь введет не то, что от него ожидает программа, она сообщит ему об этом и попросит ввести число повторно.

При вводе единицы, программа вызывает функцию work_with_console(), которая считывает массив из консоли и передает его в функцию sort_v().

При вводе двойки, программа спрашивает у пользователя имя файла, в котором хранится массив для считывания, считывает его и передает в функцию sort_v().

Функция sort_v() получает на вход массив и его длину. Если длина массива равна единице, то функция не трогает этот массив. Если длина массива равна двум, то функция сортирует этот массив по возрастанию. Если длина массива более двух, то функция делит его на две части так, чтобы длина первой части равнялась второй, либо первая часть была длиннее второй на 1, после чего функция вызывает себя же для каждого из двух новых массивов.

3. ТЕСТИРОВАНИЕ

3.1 ПРОЦЕСС ТЕСТИРОВАНИЯ

Программа собрана в операционной системе Ubuntu 18.04.1 LTS bionic компилятором g++ (Ubuntu 7.3.0-16ubuntu3) 7.3.0. В других ОС и компиляторах тестирование не проводилось.

3.2 РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

Тестовые случаи представлены в Приложении А.

Результаты тестирования показали, что программа работает верно, значит поставленная задача выполнена.

4. ВЫВОД

В ходе выполнения данной работы были изучены основные принципы рекурсивного программирования. Была создана программа реализующая рекурсивную функцию Ф на языке C++, преобразующую целочисленный вектор α так что:

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

5. ПРИЛОЖЕНИЕ А: ТЕСТОВЫЕ СЛУЧАИ

№	Ввод	Вывод
1	5	5
2	2 4	2 4
3	5 3	3 5
4	1 2 3	1 2 3
5	7 4 2	4 7 2
6	45 34 56 34	34 45 34 56
7	9 8 7 6 5	8 9 7 5 6
8	23 34 16 15 14 51 19 3 2	23 34 16 14 15 19 51 2 3

6. ПРИЛОЖЕНИЕ Б: КОД ПРОГРАММЫ

```
#include <iostream>
#include <fstream>
#include <sstream>
using namespace std;

void pr_menu(){
    cout << endl << "Выберите действие:" << endl;
    cout << "1 - ввод с клавиатуры" << endl;
    cout << "2 - ввод из файла" << endl;
    cout << "0 - выход" << endl;
}

int sort_v(int *arr, int n){
    if (n == 1){
        return 0;
    }
    if (n == 2)
        if ((*arr - *(arr+1)) > 0){
            int i = *arr;
            *arr = *(arr+1);
            *(arr+1) = i;
            return 0;
        } else
        return 0;
    int n1 = n/2;
    if(2*n1 == n){
        sort_v(arr, n1);
        sort_v(arr+n1, n1);
    }
    else{
        sort_v(arr, n-n1);
        sort_v(arr+n1+1, n1);
    }
}

void work_with_console(){
    int n;
    int *arr = new int[n];
    cout << "введите количество элементов массива" << endl;
    cin >> n;
    if (n<=0){
        cout << "неверное количество" << endl;
        return;
    }
    cout << "введите массив" << endl;
    for(int i = 0; i < n; i++)
        cin >> *(arr+i);
    sort_v(arr, n);
    for(int i = 0; i < n; i++)
        cout << *(arr+i) << " ";
}

void work_with_file(){
```

```
    string file_name;
    int *arr_f = new int[200];
    int tmp = 0;
    cout << "Введите имя файла" << endl;
    cin >> file_name;
    ifstream f;
    f.open(file_name.c_str());
    if (!f){
        cout << "Файл не открыт!" << endl;
        return;
    }
    for(f >> *(arr_f+tmp); !f.eof(); ++tmp, f >> *(arr_f+tmp));
    sort_v(arr_f, tmp+1);
    for(int i = 0; i < tmp+1; i++)
        cout << *(arr_f+i) << " ";
    delete(arr_f);
    f.close();
}

int main(){
    pr_menu();
    int way = 0;
    cin >> way;
    while(way){
        switch (way){
            case 1:
                work_with_console();
                cout << endl;
                pr_menu();
                cin >> way;
                break;
            case 2:
                work_with_file();
                cout << endl;
                pr_menu();
                cin >> way;
                break;
            default:
                cout << "Неверно введены данные!" << endl;
                pr_menu();
                cin >> way;

        }
    }
    return 0;
}
```