

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра математического обеспечения и применения ЭВМ**

**ОТЧЕТ**  
**по лабораторная работа №1**  
**по дисциплине «Алгоритмы и структуры данных»**  
**ТЕМА: ПРОГРАММИРОВАНИЕ РЕКУРСИВНЫХ АЛГОРИТМОВ**

Студент гр. 9303

\_\_\_\_\_

Павлов Д.Р.

Преподаватель

\_\_\_\_\_

Филатов А.Ю.

Санкт-Петербург

2020

### Цель работы.

Ознакомиться с основными понятиями и приемами рекурсивного программирования. Научиться решать задачи при помощи рекурсивных функций.

### Задание.

Вариант 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.

### Основные теоретические положения.

Рекурсивным называется объект, содержащий сам себя или определенный с помощью самого себя.

Мощность рекурсии связана с тем, что она позволяет определить бесконечное множество объектов с помощью конечного высказывания. Точно так же бесконечные вычисления можно описать с помощью конечной рекурсивной программы. Рекурсивные алгоритмы лучше всего использовать, когда решаемая задача, вычисляемая функция или обрабатываемая структура данных определены с помощью рекурсии.

Если процедура (функция)  $P$  содержит явное обращение к самой себе, она называется прямо рекурсивной. Если  $P$  содержит обращение к процедуре (функции)  $Q$ , которая содержит (прямо или косвенно) обращение к  $P$ , то  $P$  называется косвенно рекурсивной.

## **Выполнение работы.**

Создаем структуру вектора `struct vector`, которая включает в себя массив и размер массива.

Далее создаем функцию `transform`, которая принимает массив, начальный элемент и длину вектора. Затем функция преобразует вектора, исходя из его нормы. Потом создаем функцию `int main`, в которой распечатываем преобразованный вектор. Также в функции `main` мы проверяем что введено было именно число.

Разработанный программный код см. в приложении А.

Результаты тестирования см. в приложении Б.

## **Вывод.**

Было изучено понятие рекурсия.

Была написана программа, которая преобразует вектор, по тому как определена функция. Также программа считывает данные из клавиатуры, которые представляют вектор. В итоге программа выводит преобразованный целочисленный вектор.

# ПРИЛОЖЕНИЕ А

## ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.cpp

```
#include <iostream>

struct vector{
    int* arr;
    int size;
};

void transform(vector &v,int start, int len){
    if(len == 1) return; //alpha

    if(len == 2 &&(v.arr[start+1] < v.arr[start])){ //swap ba
        int temp = v.arr[start];
        v.arr[start] = v.arr[start+1];
        v.arr[start+1] = temp;
    }

    if(len > 2){
        //Transform(B)Transform(g)
        if(len%2 == 0){
            transform(v, start, len/2);
            transform(v, start + len/2, len/2);
        }else{
            transform(v, start, len/2+1);
            transform(v, start+len/2, len/2);
        }
    }
}

int main(){
    vector v;
    char str[10];
    std::cout<<"Enter the length of the string: ";
    fgets(str, sizeof(str), stdin);
    while (sscanf(str, "%d", &v.size) != 1) {
        std::cout<<"ERROR!\nTry to input a number:";
        fgets(str, sizeof(str), stdin);
    }

    int tempArr[v.size];
    v.arr = tempArr;
    for (int i=0;i<v.size;i++){
        std::cout<<"Number "<<i+1<<": ";
        fgets(str, sizeof(str), stdin);
        while (sscanf(str, "%d", &v.arr[i]) != 1) {
            std::cout<<"ERROR!\nTry to input a number:";
            fgets(str, sizeof(str), stdin);
        }
    }

    transform(v, 0, v.size);
    std::cout<<"Transformed vector: (";
    for (size_t i = 0; i < v.size; i++)
    {
        if(i== v.size -1)
            std::cout<<v.arr[i];
        else
            std::cout<<v.arr[i]<<",";
    }
}
```

```
    }  
    std::cout<<" "<<'\\n';  
    return 0;  
}
```

## ПРИЛОЖЕНИЕ Б

### ТЕСТИРОВАНИЕ

Таблица 1 – Результаты тестирования.

№	Входные данные	Выходные данные	Комментарии
1	Enter the length of the string: 5  Number 1: 1  Number 2: 2  Number 3: 3  Number 4: 4  Number 5: 5	Transformed vector: (1,2,3,4,5)	Программа работает  корректно
2	Enter the length of the string: 4  Number 1: 4  Number 2: 3  Number 3: 2  Number 4: 1	Transformed vector: (3,4,1,2)	Программа работает  корректно
3	Enter the length of the string: 2  Number 1: 1  Number 2: 2	Transformed vector: (1,2)	Программа работает  корректно
4	Enter the length of the string: 2  Number 1: 2  Number 2: 1	Transformed vector: (1,2)	Программа работает  корректно

