

**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. 9303

\_\_\_\_\_

Алексеевко Б.

Преподаватель

\_\_\_\_\_

Филатов А.Ю.

Санкт-Петербург

2020

## Цель работы.

Ознакомится с понятием рекурсии и научиться пользоваться алгоритмами, которые, в свою очередь используют рекурсию. Написать программу, которая будет вычислять биномиальные коэффициенты при помощи рекуррентной формулы.

## Задание.

### Вариант 1

1. Для заданных неотрицательных целых  $n$  и  $m$  вычислить (рекурсивно) биномиальные коэффициенты, пользуясь их определением:

$$C_n^m = \begin{cases} 1, & \text{если } m = 0, n > 0 \text{ или } m = n \geq 0, \\ 0, & \text{если } m > n \geq 0, \\ C_{n-1}^{m-1} + C_{n-1}^m & \text{в остальных случаях.} \end{cases}$$

## Основные теоретические положения.

Рекурсия — это процесс, определение, изображение или что-либо, которое является частью самого себя.

В программировании рекурсия — это вызов функции из нее же. Количество вложенных вызовов функции или процедуры называется глубиной рекурсии. Рекурсивная программа позволяет описать повторяющееся или даже потенциально бесконечное вычисление, причём без явных повторений частей программы и использования циклов. Рекурсия так же используется во многих алгоритмах.

Рекурсивные функции используют стек вызовов. Когда программа вызывает функцию, функция отправляется на верх стека вызовов. Адрес возврата и локальные переменные функции записываются в стек, благодаря чему каждый следующий рекурсивный вызов этой функции пользуется своим набором локальных переменных и за счёт этого работает корректно. Обратной

стороной этого механизма является то, что при очень большой глубине рекурсии может наступить переполнение стека вызовов.

### Выполнение работы.

В ходе выполнения лабораторной работы была реализована функция `binomical`, которая, в соответствии с рекуррентной формулой высчитывает биномиальный коэффициент. Для того чтобы лучше понять принцип рекуррентной формулы можно представить треугольник Паскаля:

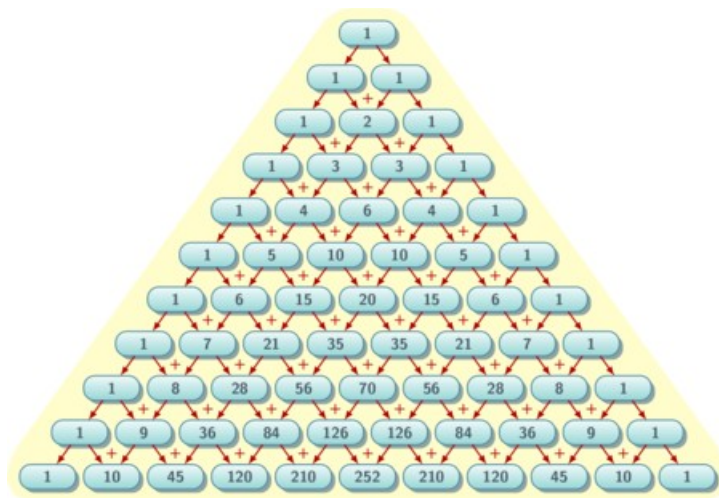


Рисунок 1 — Треугольник Паскаля.

Для вычисления  $n$ -го элемента нужно посчитать сумму элементов, стоящих над ним, в тоже время, чтобы вычислить эти два элемента нужно посчитать сумму элементов, стоящих над ними и т. д.

В функции `main()` запрашивается имя файла с входными значениями, после они передаются в функцию `binomical`.

### Выводы.

В процессе выполнения лабораторной работы было изучено понятие рекурсия, её приёмы и особенности. Была реализована программа, которая использует рекурсию для вычисления биномиального коэффициента.

## ПРИЛОЖЕНИЕ

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Файл main.cpp

```
#include <iostream>
#include <fstream>
using namespace std;
#define ull unsigned long long

ull binomical(ull m, ull n, int& depth) {
    depth += 1;
    if ((m == 0 && n > 0) || (m == n && n >= 0)) return 1;
    if (m > n && n >= 0) return 0;
    return binomical(m - 1, n - 1, depth) + binomical(m, n - 1, depth);
}

int main()
{
    string fname;
    cout << "Enter input file's name: ";
    cin >> fname;
    ifstream fin(fname);
    if (!fin.is_open()) {
        cout << "ERROR";
        return 0;
    }
    ofstream fout("out.txt");
    ull n;
    ull m;
    string str;
    while(1) {
        fin >> n >> m ;
        if (fin.eof()){
            break;
        }
        int depth = 0;
        fout << endl << binomical(n, m, depth) << endl;
        fout << "Depth: " << depth << endl;
    }
    fin.close();
    fout.close();
    return 0;
}
```

## ПРИЛОЖЕНИЕ Б

### ТЕСТИРОВАНИЕ

Входные данные	Выходные данные	Комментарий
1. 3 7	35 Depth: 69	Программа работает корректно

2. 5 15 2 3 4 9	3003 Depth: 6005  3 Depth: 5  126 Depth: 251	Программа работает коррект- но
-----------------------	---	-----------------------------------