

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

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

Студент гр. 7381

Адамов Я.В.

Преподаватель

Фирсов М.А.

Санкт-Петербург

2018

Цель работы.

Ознакомиться с рекурсией и научиться применять её на практике, написав программу, которая использует рекурсию.

Основные теоретические положения.

Рекурсия — определение, описание, изображение какого-либо объекта или процесса внутри самого этого объекта или процесса, то есть ситуация, когда объект является частью самого себя. В программировании рекурсией называется процедура, когда функция вызывает саму себя.

Задание.

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

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

Ход работы.

Программа написана на языке C.

Исходный файл: main.c

В начале работы происходит ввод данных (коэффициенты m и n), после чего происходит проверка на корректность. Если введённые данные некорректны, то программа выводит соответствующую ошибку и завершает работу. В случае ввода корректных данных происходит дальнейший вызов функции `binominal_coef()`.

```
int binominal_coef(int m, int n, int level):
```

функция принимает три целочисленных значения: коэффициенты n и m , а также значение $level$, которое обозначает глубину рекурсии. Если $m > n$, функция возвращает 0, если $m = 0$ или $m = n$, функция возвращает 1, в остальных случаях происходит рекурсивный вызов функции $C(m, n) = C(m-1, n-1) + C(m, n-1)$. О каждом вызове функции сообщается пользователю, причём чем глубже рекурсия, тем больший отступ имеет сообщение. Пример для $m = 2, n = 4$:

call: $m = 2, n = 4$; return $C(1,3) + C(2,3)$

call: $m = 1, n = 3$; return $C(0,2) + C(1,2)$

call: $m = 0, n = 2$; return 1

call: $m = 1, n = 2$; return $C(0,1) + C(1,1)$

call: $m = 0, n = 1$; return 1

call: $m = 1, n = 1$; return 1

call: $m = 2, n = 3$; return $C(1,2) + C(2,2)$

call: $m = 1, n = 2$; return $C(0,1) + C(1,1)$

call: $m = 0, n = 1$; return 1

call: $m = 1, n = 1$; return 1

call: $m = 2, n = 2$; return 1

В конце функция возвращает целочисленное значение, которое и является биномиальным коэффициентом. Это значение выводится на экран, после чего программа завершает свою работу.

Для демонстрации работы было написано несколько тестов, а также скрипт `perform_tests.sh`, который запускает все эти тесты.

Тестирование программы.

Было создано несколько тестов для проверки работы программы. Помимо тестов, демонстрирующих работу алгоритма, были написаны тесты, содержащие некорректные данные: ввод отрицательных чисел; значение m больше n ; ввод не целых чисел — демонстрирующие, как программа сообщает пользователю об ошибках ввода.

Вывод.

В ходе выполнения работы были получены навыки работы с рекурсией.

Код программы.

```
#include
<stdio.h>

// сообщения об ошибках
void error_message(int numb_of_error){
    printf("\nОшибка: ");
    switch (numb_of_error){
        case 1:
            printf("вы ввели отрицательное значение.\n");
            break;
        case 2:
            printf("m > n\n");
            break;
        case 3:
            printf("необходима ввести целое число.\n");
            break;
    }
    printf("Программа завершила работу.\n\n");
}

// поиск значения биномиального коэффициента рекурсивным способом
int binominal_coef(int m , int n, int level){
    int i;
    for(i = 0; i < level; i++) // создание отступов
        printf("    ");
    printf("call: m = %d, n = %d ; ", m, n);
    if (m == 0 || m == n){
        printf("return 1\n");
        return 1;
    } else if (m > n){
        printf("return 0\n");
        return 1;
    }
```

```

    } else {
        printf("return C(%d,%d) + C(%d,%d)\n", m-1, n-1, m, n-1);
        return binominal_coef(m-1, n-1, level+1) + binominal_coef(m, n-1, level+1);
    }
}

```

```

int main(void){

    int m, n, result;

    // описание программы
    printf("\nПрограмма для вычисления биномиальных коэффициентов.\n");
    printf("Вычисление происходит рекурсивно.\n");
    printf("\n m | 1, если m = 0, n = 0 или m = n >= 0, \n");
    printf("C   = | 0, если m > n >= 0, \n");
    printf(" n | C(m-1, n-1) + C(m, n-1) в остальных случаях.\n");
    printf("\nЗначения m и n - целые неотрицательные числа.\n\n");

    // ввод данных
    printf("Введите значение m : ");
    if (scanf("%d", &m) == 0){
        error_message(3);
        return 0;
    }
    printf("Введите значение n : ");
    if (scanf("%d", &n) == 0){
        error_message(3);
        return 0;
    }
    if (m < 0 || n < 0){
        error_message(1);
        return 0;
    }
    if (m > n){
        error_message(2);
        return 0;
    }

    // вывод результата

```

```
printf("\nХод работы алгоритма:\n\n");
result = binominal_coef(m, n, 1);
printf("\n  %d   \n", m);
printf("C    = %d\n", result);
printf("  %d   \n", n);
printf("\nПрограмма завершила работу.\n\n");

return 0;
}
```