

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.

Вариант №1.

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

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

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 coef_m, int coef_n, int level):`

Параметры:

- **coef_m**: коэффициент m.
- **coef_n**: коэффициент n;
- **level**: глубина рекурсии

Возвращаемое значение:

Биномиальный коэффициент из n по m.

Описание:

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

$$C(1, 3) + C(2, 3) = 6$$

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; ввод не целых чисел — демонстрирующие, как программа сообщает пользователю об ошибках ввода (см. Приложение А).

Вывод.

В ходе выполнения работы были получены навыки работы с рекурсией.

Приложение А. Тестирование.

Test 1:

input: $m = 2, n = 3$

output:

Программа для вычисления биномиальных коэффициентов.

Вычисление происходит рекурсивно.

$m \quad | \quad 1$, если $m = 0$, $n = 0$ или $m = n \geq 0$,
 $C \quad = \quad | \quad 0$, если $m > n \geq 0$,
 $n \quad | \quad C(m-1, n-1) + C(m, n-1)$ в остальных случаях.

Значения m и n – целые неотрицательные числа.

Введите значение m : Введите значение n :

Ход работы алгоритма:

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

2
C = 3
3

Программа завершила работу.

Test 2:

input: $m = 3, n = 5$

output:

Программа для вычисления биномиальных коэффициентов.

Вычисление происходит рекурсивно.

```
m | 1, если m = 0, n = 0 или m = n >= 0,
C = | 0, если m > n >= 0,
n | C(m-1, n-1) + C(m, n-1) в остальных случаях.
```

Значения m и n – целые неотрицательные числа.

Введите значение m : Введите значение n :

Ход работы алгоритма:

```
call: m = 3, n = 5 ; return C(2,4) + C(3,4)
  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
      C(0,1) + C(1,1) = 2
    C(0,2) + C(1,2) = 3
  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
    C(0,1) + C(1,1) = 2
  call: m = 2, n = 2 ; return 1
  C(1,2) + C(2,2) = 3
C(1,3) + C(2,3) = 6
call: m = 3, n = 4 ; return C(2,3) + C(3,3)
  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
    C(0,1) + C(1,1) = 2
  call: m = 2, n = 2 ; return 1
  C(1,2) + C(2,2) = 3
  call: m = 3, n = 3 ; return 1
  C(2,3) + C(3,3) = 4
C(2,4) + C(3,4) = 10
```

3

C = 10

Программа завершила работу.

Test 3:

input: $m = 1, n = 7$

output:

Программа для вычисления биномиальных коэффициентов.

Вычисление происходит рекурсивно.

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

Значения m и n – целые неотрицательные числа.

Введите значение m : Введите значение n :

Ход работы алгоритма:

```

call: m = 1, n = 7 ; return C(0,6) + C(1,6)
  call: m = 0, n = 6 ; return 1
  call: m = 1, n = 6 ; return C(0,5) + C(1,5)
    call: m = 0, n = 5 ; return 1
    call: m = 1, n = 5 ; return C(0,4) + C(1,4)
      call: m = 0, n = 4 ; return 1
      call: m = 1, n = 4 ; return C(0,3) + C(1,3)
        call: m = 0, n = 3 ; return 1
        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
C(0,1) + C(1,1) = 2
C(0,2) + C(1,2) = 3
C(0,3) + C(1,3) = 4
C(0,4) + C(1,4) = 5
C(0,5) + C(1,5) = 6
C(0,6) + C(1,6) = 7

```

```

1
C   = 7
7

```

Программа завершила работу.

Test 4:

input: m = 5, n = 2

output:

Программа для вычисления биномиальных коэффициентов.
Вычисление происходит рекурсивно.

```

m   | 1, если m = 0, n = 0 или m = n >= 0,
C   = | 0, если m > n >= 0,
n   | C(m-1, n-1) + C(m, n-1) в остальных случаях.

```

Значения m и n – целые неотрицательные числа.

Введите значение m : Введите значение n :

Ход работы алгоритма:

```

call: m = 5, n = 2 ; return 0

```


$$C_2 = 0$$

Программа завершила работу.

Test 5:

input m = -4, n = 6

output:

Программа для вычисления биномиальных коэффициентов.
Вычисление происходит рекурсивно.

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

Значения m и n - целые неотрицательные числа.

Введите значение m : Введите значение n :

Ошибка: вы ввели отрицательное значение.

Программа завершила работу.

Test 6:

input m = ksj, n = 10

output:

Программа для вычисления биномиальных коэффициентов.
Вычисление происходит рекурсивно.

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

Значения m и n – целые неотрицательные числа.

Введите значение m :

Ошибка: необходима ввести целое число.

Программа завершила работу.

Приложение Б. 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("необходима ввести целое число.\n");
            break;
    }
    printf("Программа завершила работу.\n\n");
}

// поиск значения биномиального коэффициента рекурсивным способом
int binominal_coef(int coef_m , int coef_n, int level){
    for(int i = 0; i < level; i++) // создание отступов
        printf("    ");
    printf("call: m = %d, n = %d ; ", coef_m, coef_n); // сообщение о вызове функции
    if ( (coef_m == 0 && coef_n > 0) || (coef_m == coef_n && coef_n >= 0) ){
        printf("return 1\n"); // самый глубокий уровень, возврат единицы
        return 1;
    } else if (coef_m > coef_n && coef_n >= 0){
        printf("return 0\n"); // случай, возможный только при вводе  $m > n$ 
        return 0;
    }
}
```

```

    } else {
        // вызов рекурсии в случае, если m < n
        printf("return C(%d,%d) + C(%d,%d)\n", coef_m-1, coef_n-1, coef_m, coef_n-1);
        int result = binominal_coef(coef_m-1, coef_n-1, level+1) + binominal_coef(coef_m,
coef_n-1, level+1);
        for(int i = 0; i < level; i++)
            printf(" ");
        printf("C(%d,%d) + C(%d,%d) = %d\n", coef_m-1, coef_n-1, coef_m, coef_n-1,
result);
        return result;
    }
}

```

```

int main(void){

    int coef_m, coef_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", &coef_m) == 0){ // ввод и проверка на корректность коэффициента m
        error_message(2);
        return 0;
    }
    printf("Введите значение n : ");
    if (scanf("%d", &coef_n) == 0){ // ввод и проверка на корректность коэффициента n
        error_message(2);
        return 0;
    }
    if (coef_m < 0 || coef_n < 0){ // проверка на отсутствие отрицательных значений
        error_message(1);
        return 0;
    }
}

```

```
// вывод результата
printf("\nХод работы алгоритма:\n\n");
result = binominal_coef(coef_m, coef_n, 1); // вызов функции, получение результата
// вывод результата
printf("\n  %d  \n", coef_m);
printf("C    = %d\n", result);
printf("  %d  \n", coef_n);
printf("\nПрограмма завершила работу.\n\n");

return 0;
}
```