

## Задание легкой сложности

Исполнитель может передвигаться по координатной плоскости.

У исполнителя есть 3 команды, которые обозначены латинскими буквами:

- A. Сделать 1 шаг вправо**
- B. Сделать 1 шаг вверх**
- C. Сделать 2 шага вправо и 3 шага вверх**

Программа для исполнителя – это последовательность команд.

Сколько существует программ, в результате которых исполнитель окажется в координате (7; 13), если изначально находился в координате (0; 0)?

Траектория движения исполнителя – это последовательность результатов выполнения всех команд программы. *Например*, для программы **ABC** при исходной координате (0; 0) траектория будет состоять из координат (1; 0), (1; 1), (3; 4).

## Решение задания легкой сложности

Используем метод рекурсивного программирования. Создадим функцию `solve(координата X, координата Y)`, которая будет возвращать, сколькими способами можно добраться до (7; 13), если исходная координата исполнителя была (X; Y). Функция будет возвращать сумму результатов из координат, куда мы можем попасть за 1 команду: `solve(X + 1, Y) + solve(X, Y + 1) + solve(X + 2, Y + 3)`. Также будем проверять, чтобы исполнитель не оказался правее или выше необходимой координаты.

Пример решения на языке python:

```
def solve(X, Y):  
    if X == 7 and Y == 13:  
        return 1  
    if X > 7 or Y > 13:  
        return 0  
    return solve(X + 1, Y) + solve(X, Y + 1) + solve(X + 2, Y + 3)  
  
print(solve(0, 0))
```