Тестова Мария Геннадьевна, https://vk.com/shchashcha
Основано на 23 задании ЕГЭ.

Задание высокой 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 (13; 12), если изначально находился в координате (1; -1), при этом программа содержит не более 15 команд и траектория движения не содержит координат, где оба числа равны друг другу. *Например* (4; 4), (10, 10) и т.д.

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 высокой сложности

Используем метод рекурсивного программирования. Создадим функцию solve(координата X, координата Y, количество выполненных команд), которая будет возвращать, сколькими способами можно добраться до (13; 12), 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 и не попал в точки с равными координатами. Если исполнитель превысит 15 команд, программа должна вернуть 0.

Пример решения на языке python:

```
def solve(X, Y, n):
    if X == 13 and Y == 12:
        return 1
    if X > 13 or Y > 12:
        return 0
    if X == Y or n >= 15:
        return 0
    return solve(X + 1, Y, n + 1) + solve(X, Y + 1, n + 1) + solve(X + 2, Y
+ 3, n + 1)

print(solve(1, -1, 0))
```