

Исполнитель преобразует число на экране. У исполнителя есть три команды, которым присвоены номера:

1. Прибавить 1

2. Прибавить 2

3. Умножить на 3

Первая команда увеличивает число на экране на 1, вторая увеличивает его на 2, третья — умножает на 3.

Программа для исполнителя — это последовательность команд. Сколько существует программ, которые преобразуют исходное число 1 в число 15, и при этом траектория вычислений содержит число 10?

Траектория вычислений — это последовательность результатов выполнения всех команд программы. Например, для программы 231 при исходном числе 4 траектория будет состоять из чисел 6, 18, 19.

Решение.

Пусть $R(n)$ — количество программ, которые число 1 преобразуют в число n .

Верны следующие соотношения:

$$R(n) = R(n-1) + R(n-2) + R(n/3) \text{ (если } n \text{ — кратно 3)}.$$

Также будем учитывать то, что траектория вычислений должна содержать число 10.

$$R(1) = 1.$$

$$R(2) = R(1) = 1.$$

$$R(3) = R(2) + R(1) + R(1) = 3.$$

$$R(4) = R(3) + R(2) = 4.$$

$$R(5) = R(4) + R(3) = 7.$$

$$R(6) = R(5) + R(4) + R(2) = 12.$$

$$R(7) = R(6) + R(5) = 19.$$

$$R(8) = R(7) + R(6) = 31.$$

$$R(9) = R(8) + R(7) + R(3) = 53.$$

$$R(10) = R(9) + R(8) = 84.$$

$$R(11) = R(10) = 84.$$

$$R(12) = R(11) + R(10) = 168.$$

$$R(13) = R(12) + R(11) = 252.$$

$$R(14) = R(13) + R(12) = 420.$$

$$R(15) = R(14) + R(13) = 672.$$

Таким образом, количество программ, удовлетворяющих условию задачи, равно 672.

Ответ: 672.

Приведём другое решение на языке Python.

```
def f(x, y):
    if x > y:
        return 0
    if x == y:
        return 1
    else:
        return f(x + 1, y) + f(x + 2, y) + f(x * 3, y)
print(f(1, 10) * f(10, 15))
```