

Исполнитель преобразует число на экране.

У исполнителя есть две команды, которым присвоены номера:

1. Прибавить 1

2. Умножить на 3

Первая команда увеличивает число на экране на 1, вторая умножает его на 3. Программа для исполнителя — это последовательность команд. Сколько существует программ, которые преобразуют исходное число 1 в число 65, и при этом траектория вычислений содержит число 20?

Траектория вычислений — это последовательность результатов выполнения всех команд программы. Например, для программы 212 при исходном числе 4 траектория будет состоять из чисел 12, 13, 39.

Решение.

Искомое количество программ равно произведению количества программ, получающих из числа 1 число 20, на количество программ, получающих из числа 20 число 65, при этом траектория вычислений должна содержать число 20.

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

Верны следующие соотношения:

$R(n) = R(n-1) + R(n/3)$ (если n — кратно 3).

$R(1) = 1.$

$R(2) = 1$

$R(3) = 2.$

$R(4) = 2.$

$R(5) = 2.$

$R(6) = 3.$

$R(7) = 3.$

$R(8) = 3.$

$R(9) = 5.$

$R(10) = 5.$

$R(11) = 5.$

$R(12) = 7.$

$R(13) = 7.$

$R(14) = 7.$

$R(15) = 9.$

$R(16) = 9.$

$R(17) = 9.$

$R(18) = 12.$

$R(19) = 12.$

$R(20) = 12.$

Из числа 20 число 65 можно получить тремя способами: последовательностью команд 211111, последовательностью команд 1211 и последовательностью команд 11..11.

Таким образом, количество программ, удовлетворяющих условию задачи, равно $12 \cdot 3 = 36$.

Ответ: 36.

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 * 3, y)
print(f(1, 20) * f(20, 65))
```