

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

1. Прибавить 1
2. Прибавить 2
3. Умножить на 2

Сколько существует программ, для которых при исходном числе 4 результатом является число 13, и при этом траектория вычислений содержит число 11?

Решение.

Искомое количество программ равно произведению количества программ, получающих из числа 4 число 11, на количество программ, получающих из числа 11 число 13.

Пусть $R(n)$ — количество программ, которые число 4 преобразуют в число n , а $P(n)$ — количество программ, которые число 11 преобразуют в число n .

Для всех $n > 7$ верны следующие соотношения:

1. Если n не делится на 2, то тогда $R(n) = R(n - 1) + R(n - 2)$, так как существует два способа получения n — прибавлением единицы или прибавлением двойки. Аналогично $P(n) = P(n - 1) + P(n - 2)$

2. Если n делится на 2, тогда $R(n) = R(n - 1) + R(n - 2) + R(n / 2)$. Аналогично $P(n) = P(n - 1) + P(n - 2) + P(n / 2)$

Последовательно вычислим значения $R(n)$:

$$R(4) = 1$$

$$R(5) = R(4) = 1$$

$$R(6) = R(4) + R(5) = 1 + 1 = 2$$

$$R(7) = R(5) + R(6) = 2 + 1 = 3$$

$$R(8) = R(6) + R(7) + R(4) = 2 + 3 + 1 = 6$$

$$R(9) = R(7) + R(8) = 3 + 6 = 9$$

$$R(10) = R(8) + R(9) + R(5) = 6 + 9 + 1 = 16$$

$$R(11) = R(9) + R(10) = 9 + 16 = 25$$

Теперь вычислим значения $P(n)$:

$$P(11) = 1$$

$$P(12) = P(11) = 1$$

$$P(13) = P(11) + P(12) = 2$$

Таким образом, количество программ, удовлетворяющих условию задачи, равно $25 \cdot 2 = 50$.

Ответ: 50.

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 * 2, y)
print(f(4, 11) * f(11, 13))
```