

С обязательным этапом – программа должна обязательно пройти через определённое число

Исполнитель преобразует число, записанное на экране, с помощью следующих команд:

- А – прибавить 1;
- В – прибавить 2;
- С – умножить на 2.

Программа для исполнителя – это последовательность команд, записанных в виде букв. При выполнении программы от исходного числа получается траектория вычислений – последовательность чисел, полученных после выполнения каждой команды.

Найдите количество программ, которые преобразуют исходное число **4** в число **15**, при этом траектория вычислений программы должна содержать оба числа – **11** и **13** (числа могут появиться в любом порядке, но оба обязательно должны встречаться).

Например, для программы АСВ при исходном числе 7 траектория будет состоять из чисел 8, 16, 18.

Для эффективного решения задачи с условиями на промежуточные значения удобно использовать структурированный подход. Разберём пошаговый алгоритм:

Шаг 1. Анализ задачи. Наша цель – определить количество программ (последовательностей команд), которые преобразуют число 4 в число 15 при условии, что промежуточные результаты обязательно включают числа 11 и 13. Для этого удобно разбить процесс преобразования на три независимых этапа:

1. Преобразование из 4 в 11.
2. Преобразование из 11 в 13.
3. Преобразование из 13 в 15.

Общее число допустимых программ равно произведению числа вариантов для каждого этапа.

Шаг 2. Формулирование подзадачи. Чтобы найти число способов преобразования одного числа в другое (например, из x в y), можно использовать рекурсивный метод. Для этого вводим функцию $f(x, y)$, возвращающую количество последовательностей команд, преобразующих x в y с использованием следующих операций:

- А: прибавить 1 (то есть $x + 1$);
- В: прибавить 2 (то есть $x + 2$);

- С: умножить на 2 (то есть $x * 2$).

Шаг 3. Рекурсивное решение подзадачи.

Шаг 3.1. Базовые случаи рекурсии:

- if $x > y$: Текущее число уже превысило целевое, а дальнейшие операции только увеличат его, значит, данный путь не может привести к решению. Возвращаем 0.
- if $x == y$: Мы достигли целевого числа - найден корректный путь (последовательность команд, приведшая к у). Возвращаем 1.

Шаг 3.2. Рекурсивное разбиение. Если x меньше y , то для каждого шага выбираем одну из трёх возможных операций:

- Применить команду А: перейти от x к $x + 1$.
- Применить команду В: перейти от x к $x + 2$.
- Применить команду С: перейти от x к $x * 2$.

Таким образом, общее количество способов преобразования из x в y будет равно сумме способов преобразования для каждого из вариантов: $f(x, y) = f(x + 1, y) + f(x + 2, y) + f(x * 2, y)$.

Шаг 4. Решение исходной задачи. Чтобы обеспечить наличие в траектории промежуточных чисел 11 и 13, делим задачу на три этапа:

- Вычисляем количество способов преобразования из 4 в 11: $f(4, 11)$.
- Вычисляем количество способов преобразования из 11 в 13: $f(11, 13)$.
- Вычисляем количество способов преобразования из 13 в 15: $f(13, 15)$.

Так как эти этапы независимы, итоговое число программ - это произведение: $f(4, 11) \times f(11, 13) \times f(13, 15)$.

Шаг 5. Итоговый алгоритм.

5.1. Определим рекурсивную функцию $f(x, y)$

- Если $x > y$ - возвращаем 0;
- Если $x == y$ - возвращаем 1.
- Иначе-возвращаем сумму $f(x + 1, y) + f(x + 2, y) + f(x * 2, y)$.

5.2. Вычислить количество вариантов для каждого этапа:

- Количество способов преобразования из 4 в 11: $f(4, 11)$.
- Количество способов преобразования из 11 в 13: $f(11, 13)$.
- Количество способов преобразования из 13 в 15: $f(13, 15)$.

5.3. Найти итоговое число программ путём перемножения результатов этапов: $f(4, 11) \times f(11, 13) \times f(13, 15)$.

5.4. Вывести результат.

PYTHON

```
def f(x, y):
    # Если текущее число x превышает y, дальнейшие операции не помогут достичь
    y – возвращаем 0.
    if x > y:
        return 0
    # Если достигли целевого числа y, то найден один корректный путь –
    возвращаем 1.
    if x == y:
        return 1

    # Применяем рекурсивно все три операции:
    # - прибавляем 1: f(x + 1, y)
    # - прибавляем 2: f(x + 2, y)
    # - умножаем на 2: f(x * 2, y)
    # Суммируем все варианты для получения общего числа путей от x до y.
    return f(x + 1, y) + f(x + 2, y) + f(x * 2, y)

# Вычисляем количество программ для каждого этапа
# - преобразование из 4 в 11
# - преобразование из 11 в 13
# - преобразование из 13 в 15
# Общее количество программ равно произведению вариантов для каждого этапа.
print(f(4, 11) * f(11, 13) * f(13, 15))
```

Ответ: 100.

```
def f(x, y):
    if x > y:
        return 0
    if x == y:
        return 1

    return f(x + 1, y) + f(x + 2, y) + f(x * 2, y)

print(f(4, 11) * f(11, 13) * f(13, 15))
```