

С избегаемым этапом – программа не должна проходить через указанное число

На экране отображается некоторое число, которое может быть преобразовано с помощью исполнителя. Исполнитель умеет выполнять две операции, обозначенные латинскими буквами:

- А - уменьшить число на 2;
- В - разделить число нацело на 2.

Программа исполнителя - это последовательность команд, определяющая порядок выполнения операций. Траектория вычислений-это последовательность чисел, получаемых после применения каждой команды программы.

Определите, сколько существует программ, которые преобразуют число 30 в число 8, при этом траектория вычислений не должна содержать число 18.

Например, для программы ВА при исходном числе 35 траектория будет состоять из чисел 17,15.

Так как мы преобразуем число от большего к меньшему, условие сравнения должно выглядеть как $x < y$ (то есть, если текущее число меньше целевого, дальнейшие шаги не имеют смысла).

Чтобы исключить пути, в которых встречается число 18, в рекурсивную функцию добавлена проверка: если текущее значение равно 18, такой путь считается недопустимым, и функция возвращает 0.

def f(x, y):

 # Если текущее число меньше целевого ($x < y$) или равно 18,

 # путь считается недопустимым – возвращаем 0.

if x < y or x == 18:

return 0

 # Если текущее число совпадает с целевым, найден допустимый путь – возвращаем 1.

if x == y:

return 1

 # Рекурсивно пробуем две операции:

 # 1. Уменьшение на 2: $f(x - 2, y)$

 # 2. Целочисленное деление на 2: $f(x // 2, y)$

return f(x - 2, y) + f(x // 2, y)

Выводим количество допустимых программ, которые преобразуют число 30 в число 8.

print(f(30, 8))

Ответ: 3.

