

Задание №23 из ЕГЭ по информатике направлено на проверку умения анализировать ход выполнения алгоритма и учитывать промежуточные состояния в вычислениях. Оно относится к заданиям повышенного уровня сложности, и на его выполнение отводится около 8 минут.

Условно такие задания можно разделить на несколько типов:

1. **Базовый уровень** – подсчёт количества программ, преобразующих одно число в другое с использованием заданных команд (например: «Сколько существует программ, которые преобразуют число 10 в 30?»);
2. **С обязательным этапом** – программа должна обязательно пройти через определённое число;
3. **С избегаемым этапом** – программа не должна проходить через указанное число;
4. **С обязательным и запрещённым этапами одновременно** – траектория должна содержать одно число и не содержать другое.

23-1 (1)

Исполнитель Тренер преобразует число на экране.

У исполнителя есть две команды, которым присвоены номера.

1. **Прибавить 1.**

2. **Прибавить 2.**

Первая команда увеличивает число на экране на 1, вторая увеличивает его на 2.

Программа для исполнителя Тренер — это последовательность команд.

Сколько существует программ, для которых при исходном числе 1 результатом является число 11?

Решение.

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)  
print(f(1, 11))
```

Ответ: 89.

23-2 (4)

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

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

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

Программа для исполнителя — это последовательность команд. Например, если в начальный момент на экране находится число 1, то программа 312 последовательно преобразует его в 4, 5, 10. Сколько существует программ, которые преобразуют исходное число 3 в число 22 и при этом траектория вычислений содержит число 10 и не содержит чисел 13 и 17?

Траектория вычислений программы — это последовательность результатов выполнения всех команд программы. Например, для программы 312 при исходном числе 1 траектория будет состоять из чисел 4, 5, 10.

Решение.

Приведем решение на языке Python.

```
def fn(x, y, sp=[]):
    if x >= y:
        return 1 if x == y and 10 in sp and 13 not in sp and 17 not in sp else 0
    return fn(x+1, y, sp+[x+1]) + fn(x*2, y, sp+[x*2]) + fn(x+3, y, sp+[x+3])
print(fn(3, 22, [3]))
```

Ответ: 168.

Приведем другое решение на языке Python.

```
def f(x, y):
    if x > y or x == 13 or x == 17:
        return 0
    if x == y:
        return 1
    else:
        return f(x + 1, y) + f(x * 2, y) + f(x + 3, y)
print(f(3, 10) * f(10, 22))
```

23-3 (4)

Исполнитель Калькулятор преобразует число на экране. У исполнителя есть две команды, которые обозначены латинскими буквами.

А. вычти 3

В. подели на 2

Первая команда уменьшает число на экране на 3, вторая команда делит число на 2 (нецелый результат округляется до ближайшего целого в большую сторону). Программа для исполнителя – это последовательность команд.

Сколько существует таких программ, которые исходное число 120 преобразуют в число 20, при этом траектория вычислений не содержит чисел 35 и 51?

Траектория вычислений программы – это последовательность результатов выполнения всех команд программы.

Например, для программы АВА при исходном числе 100 траектория состоит из чисел 97, 49, 46.

Python

```
from math import *
def f(x, y):
    if x < y or x==35 or x==51:
        return 0
    if x == y:
        return 1
    if x > y:
        return f(x-3, y) + f(ceil(x/2), y)
print(f(120, 20))
```

Ответ 10