

Решение заданий №23 ЕГЭ по информатике

Задача (классическая)

У исполнителя Удвоитель две команды, которым присвоены номера:

1. прибавить 3,
2. умножить на 2.

Первая из них увеличивает число на экране на 3, вторая - удваивает его.

Программа для Удвоителя - это последовательность команд.

Сколько есть программ, которые число 1 преобразуют в число 25 ?

Число x , это то число, с которым мы работаем. Число y - это куда нужно прийти. Если число x достигло пункта назначения, то возвращаем 1. Если оно перескочило y , то возвращаем 0. А если ещё не дошло до y , то продолжаем вычисления с помощью рекурсии.

Решение:

1 способ (самый эффективный, на Python).



main.py

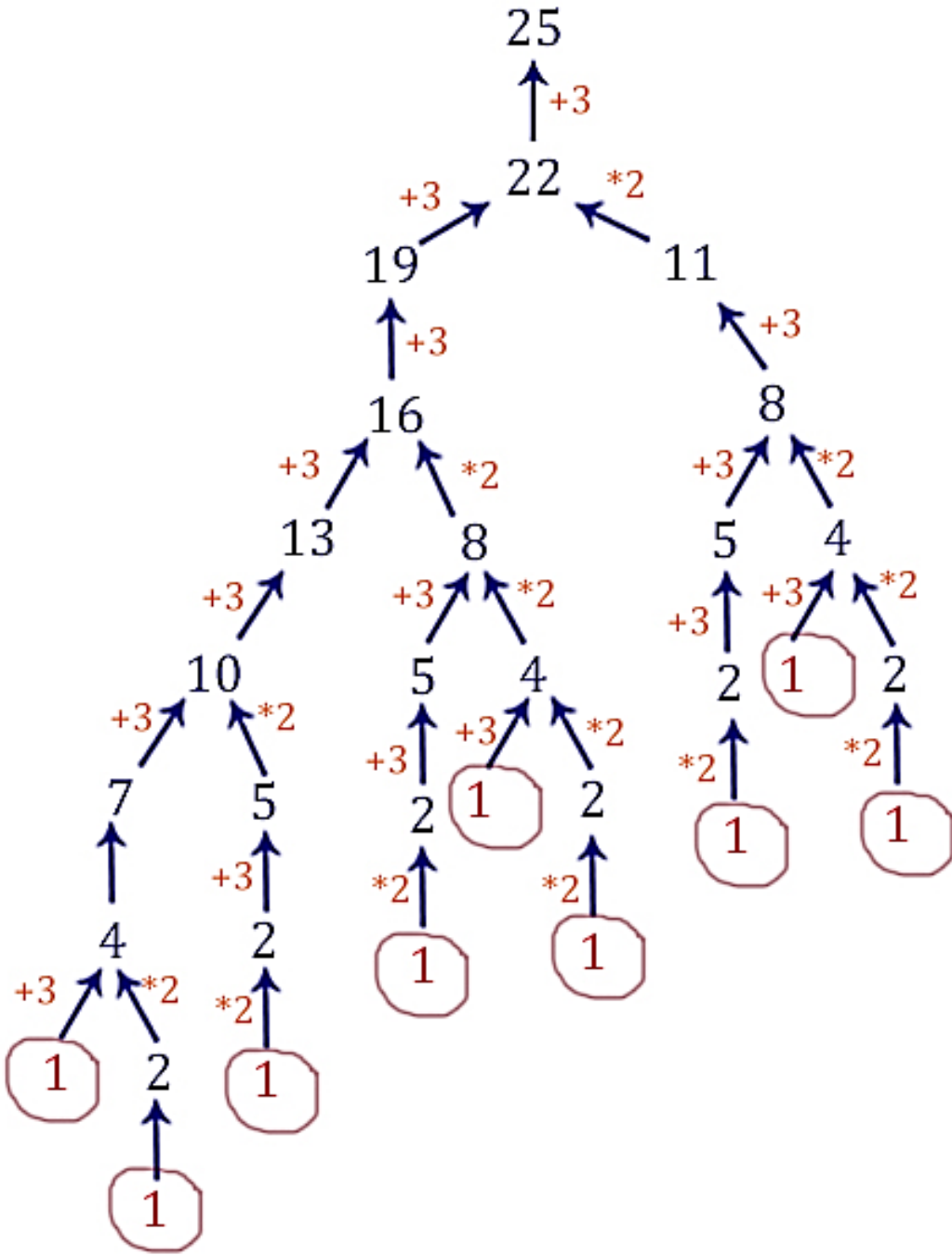
```
1 def F(x, y):
2     if x == y: return 1
3
4     if x > y: return 0
5
6     if x < y: return F(x+3, y) + F(x*2, y)
7
8 print(F(1, 25))
```

9

...Program finished with exit code 0
Press ENTER to exit console.

2 Способ (графический)

Начинаем рассматривать задачу с конца. Если число нечётное, то оно может быть получено только с помощью первой команды. Если число чётное, то оно может быть получено с помощью двух команд.



3 Способ (С помощью таблицы)

Некоторое число i можно получить только двумя способами: либо с помощью первой команды, либо с помощью второй команды. Тогда количество программ для некоторого числа i будет складываться из двух чисел: количества программ для числа $i-3$ и количества программ для числа $i / 2$ (Если i - чётное).

Числа	1	2	3	4	5	6	7	8	9	10
+3	-	-	-	1	2	3	4	5	6	7
*2	-	1	-	2	-	3	-	4	-	5
Кол. Прог.	1	1	0	2	1	0	2	3	0	3

Числа	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
+3	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
*2	-	6	-	7	-	8	-	9	-	10	-	11	-	12	-
Кол. Прог.	3	0	3	5	0	6	5	0	6	8	0	9	8	0	9

Основная идея 23 задания из ЕГЭ по информатике заключается в том, что результат каждого шага опирается на результаты предыдущих шагов!

Задача (с избегаемым узлом)

Исполнитель НечетМ преобразует число на экране. У исполнителя НечетМ две команды, которым присвоены номера:

1. прибавь 1

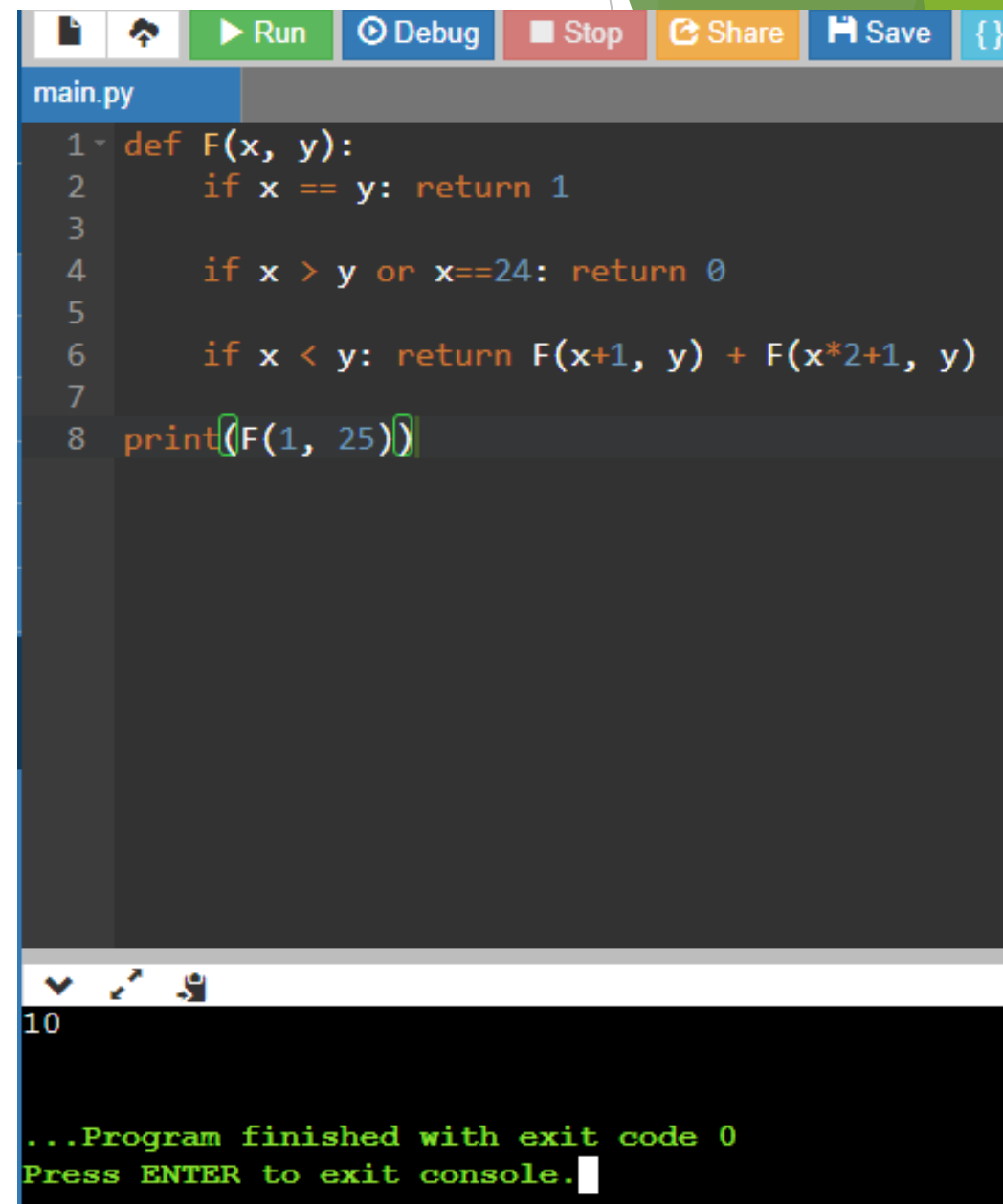
2. сделай нечётное

Первая из этих команд увеличивает число x на экране на 1, вторая переводит число x в число $2x+1$.

Например, вторая команда переводит число 10 в число 21. Программа для исполнителя НечетМ — это последовательность команд. Сколько существует таких программ, которые число 1 преобразуют в число 25, причём траектория вычислений не содержит число 24? Траектория вычислений программы — это последовательность результатов выполнения всех команд программы. Например, для программы 121 при исходном числе 7 траектория будет состоять из чисел 8, 17, 18.

Решение:

1 способ (самый эффективный, на Python).



```
main.py
1 def F(x, y):
2     if x == y: return 1
3
4     if x > y or x==24: return 0
5
6     if x < y: return F(x+1, y) + F(x*2+1, y)
7
8 print(F(1, 25))

10

...Program finished with exit code 0
Press ENTER to exit console.
```



2 Способ (графический)

3 способ (Решение с помощью таблицы).

Мы не можем получить число 24! Значит, единственным способом добраться до числа 25 будет вторая команда. Получается, что сначала нужно получить число 12, тогда $2 * 12 + 1 = 25$ ($2x+1$). Это единственный путь!

Числа	1	2	3	4	5	6	7	8	9	10	11	12
$2x+1$	-	-	1	-	2	-	3	-	4	-	5	-
$+1$	-	1	2	3	4	5	6	7	8	9	10	11
Кол. Прог.	1	1	2	2	3	3	5	5	7	7	10	10

Задача (ЕГЭ по информатике)

У исполнителя есть три команды, которым присвоены номера:

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

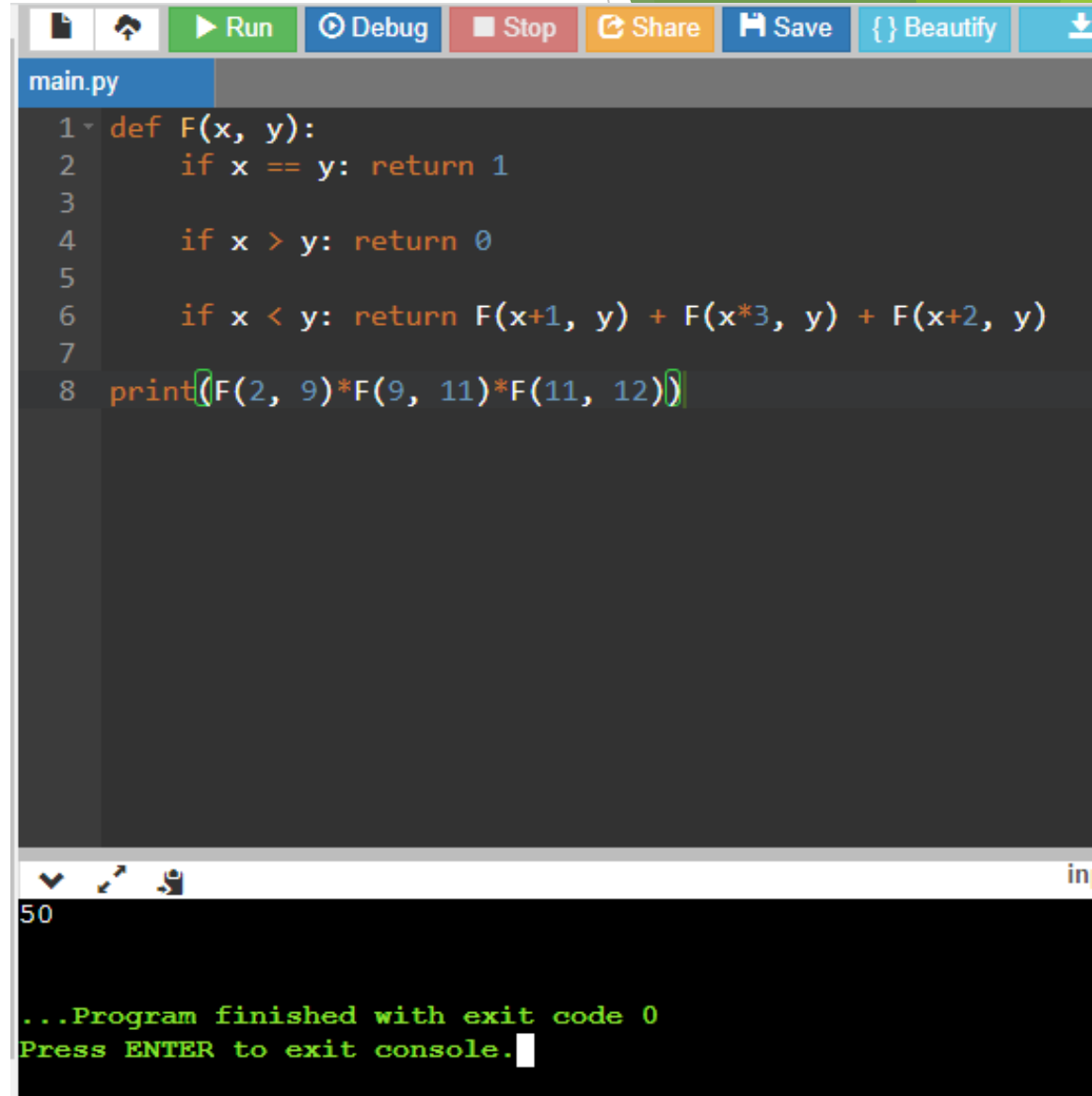
Первая команда увеличивает число на экране на 1, вторая умножает его на 3, третья увеличивает его на 2.

Сколько существует программ, которые преобразуют исходное число 2 в число 12 и при этом траектория вычислений содержит число 9 и число 11?

Траектория вычислений программы - это последовательность результатов выполнения всех команд программы. Например, для программы 132 при исходном числе 7 траектория будет состоять из чисел 8, 10, 30.

Решение:

1 способ (самый эффективный, на Python).



```
main.py
1 def F(x, y):
2     if x == y: return 1
3
4     if x > y: return 0
5
6     if x < y: return F(x+1, y) + F(x*3, y) + F(x+2, y)
7
8 print(F(2, 9)*F(9, 11)*F(11, 12))
```

50

...Program finished with exit code 0
Press ENTER to exit console.

2 способ (Решение с помощью таблицы).

От числа 11 до числа 12 можно добраться единственным путём ($11 + 1 = 12$).

От числа 9 до числа 11 можно добраться двумя способами ($9 + 1 + 1 = 11$, $9 + 2 = 11$).

Найдём сколькими способами можно попасть от числа 2 до числа 9.

Числа	2	3	4	5	6	7	8	9
+1	-	2	3	4	5	6	7	8
*3	-	-	-	-	2	-	-	3
+2	-	-	2	3	4	5	6	7
Кол-во программ	1	1	2	3	6	9	15	25

Учитывая, что от 9 до 11 двумя способами можно добраться, то $25 * 2 = 50$

Задача (ЕГЭ по информатике)

Исполнитель Май17 преобразует число на экране. У исполнителя есть две команды, которым присвоены номера:

1. Прибавить 1
2. Прибавить 3

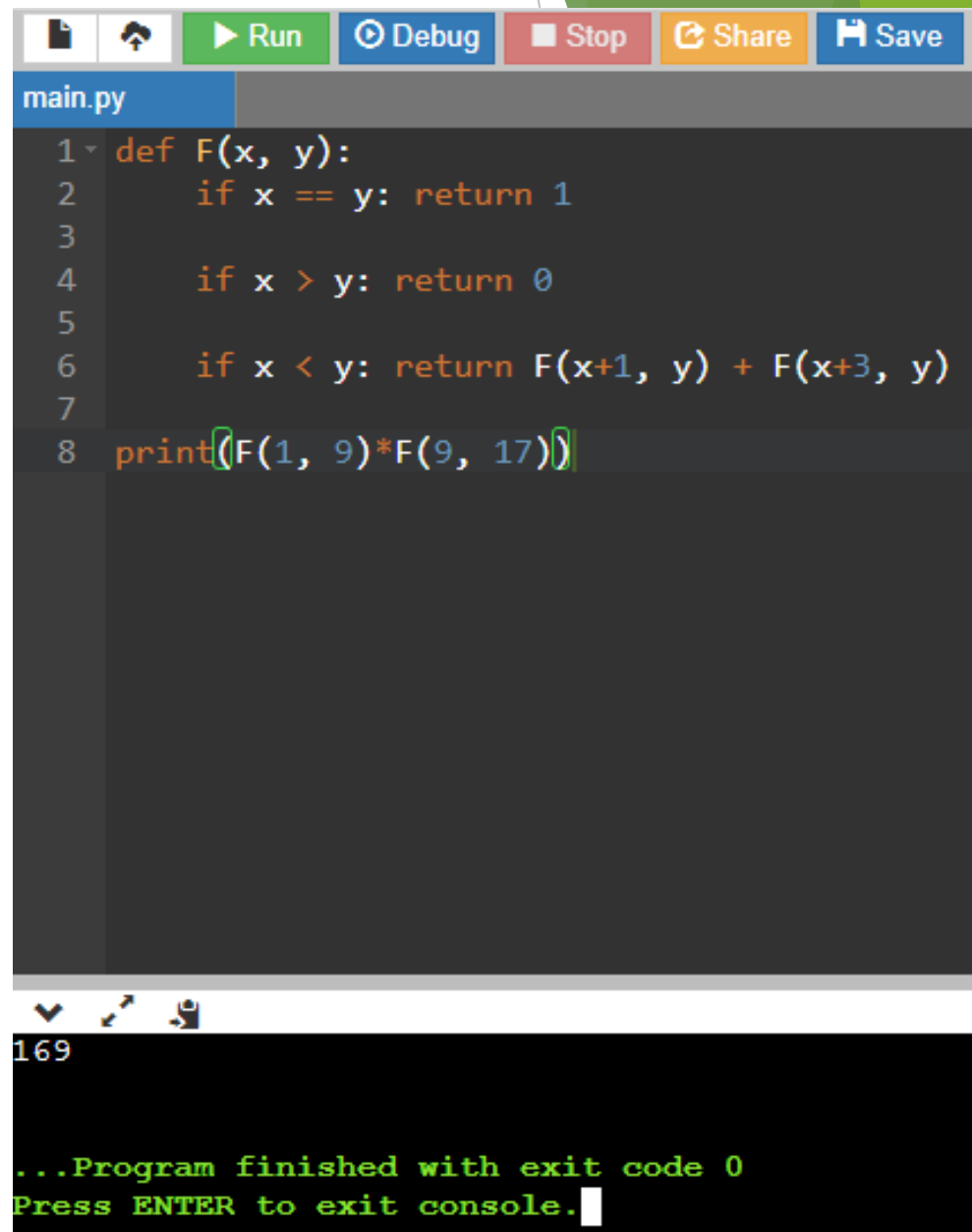
Первая команда увеличивает число на экране на 1, вторая увеличивает его на 3. Программа для исполнителя Май17 — это последовательность команд.

Сколько существует программ, для которых при исходном числе 1 результатом является число 17 и при этом траектория вычислений содержит число 9?

Траектория вычислений программы — это последовательность результатов выполнения всех команд программы. Например, для программы 12 при исходном числе 7 траектория будет состоять из чисел 8, 11, 12.

Решение:

1 способ (самый эффективный, на Python).



```
main.py
1 def F(x, y):
2     if x == y: return 1
3
4     if x > y: return 0
5
6     if x < y: return F(x+1, y) + F(x+3, y)
7
8 print(F(1, 9)*F(9, 17))

169

...Program finished with exit code 0
Press ENTER to exit console.
```

2 способ (Решение с помощью таблицы).

При составлении значения для числа 10, мы не имеем право "заглядывать" за число 9, иначе число 9 будет пропущено! Поэтому для следующих трёх чисел (9 , $9 + 1$, $9 + 1 + 1$), начиная с 9, будет 13 программ.

Числа	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
+1	-	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
+3	-	-	-	1	2	3	4	5	6	-	-	9	10	11	12	13	14
Кол-во программ	1	1	1	2	3	4	6	9	13	13	13	26	39	52	78	117	169