

## . Текстовый разбор домашки 2

### DZ\_23\_prog\_1

#### Задача № 1 (633)

У исполнителя Калькулятор три команды, которым присвоены номера:

1. прибавь 1
2. умножь на 2
3. возведи в квадрат

Сколько есть программ, которые число 5 преобразуют в число 154?

Ссылка на видео-разбор с таймингом: [https://vk.com/video-205546952\\_456241277?t=0h0m0s](https://vk.com/video-205546952_456241277?t=0h0m0s)

*Решение*

Напишем рекурсивную функцию, которая пройдет различные числовые траектории из одного числа до конечного и посчитает их количество. Функция будет принимать два параметра:

- с: текущее число.
- е: конечное число, к которому нужно прийти.

Рекурсивно она будет вычислять возможные пути от с до е.

Условия задачи:

1. Если текущее число с больше целевого числа е, то вернуть 0.
2. Если текущее число с равно целевому числу е, то вернуть 1.
3. Если текущее число с меньше целевого числа е, то вычислить возможные пути для трех действий:

- Увеличить с на 1 и рекурсивно вызвать функцию.
- Умножить с на 2 и рекурсивно вызвать функцию.
- Возвести с в квадрат и рекурсивно вызвать функцию.

Выводим на экран результат работы функции f(5,154), получаем ответ.

```
def f(c,e):  
    if c>e: return 0  
    if c==e: return 1  
    if c<e: return f(c+1,e)+f(c*2,e)+f(c**2,e)  
  
print(f(5,154))
```

Ответ: 8966

Telegram: @fast\_ege

### DZ\_23\_prog\_2

#### Задача №2 (1141)

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

1. Прибавь 1
2. Прибавь 2
3. Умножь на 3

Сколько существует программ, которые преобразуют исходное число 1 в число 15, и при этом траектория вычислений содержит число 10 и не содержит числа 14?

Ссылка на видео-разбор с таймингом: [https://vk.com/video-205546952\\_456241277?t=0h1m30s](https://vk.com/video-205546952_456241277?t=0h1m30s)

### Решение

Напишем рекурсивную функцию, которая пройдет различные числовые траектории из одного числа до конечного и посчитает их количество, при этом будем учитывать, что траектория не должна содержать число 14. Функция будет принимать два параметра:

- с: текущее число.
- е: конечное число, к которому нужно прийти.

Рекурсивно она будет вычислять возможные пути от с до е.

Условия задачи:

1. Если текущее число с больше целевого числа е, или  $s == 14$ , то вернуть 0.
2. Если текущее число с равно целевому числу е, то вернуть 1.
3. Если текущее число с меньше целевого числа е, то вычислить возможные пути для трех действий:
  - Увеличить с на 1 и рекурсивно вызвать функцию.
  - Увеличить с на 2 и рекурсивно вызвать функцию.
  - Умножить с на 3 и рекурсивно вызвать функцию.

Выводим на экран результат умножения  $f(1,10)$  – числа возможных траекторий от 1 до 10 и  $f(10,15)$  – числа возможных траекторий от 10 до 15. Таким образом мы учитываем, что траектория вычислений содержит указанное в задаче число.

```
def f(c,e):  
    if c>e or c==14: return 0  
    if c==e: return 1  
    if c<e: return f(c+1,e)+f(c+2,e)+f(c*3,e)
```

```
print(f(1,10)*f(10,15))
```

Ответ: 252

Telegram: @fast\_ege

## DZ\_23\_prog\_3

### Задача № 3 (1142)

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

1. Прибавь 1
2. Прибавь 2
3. Прибавь 3

Сколько существует программ, которые преобразуют исходное число 5 в число 18, и при этом траектория вычислений содержит число 11 и не содержит чисел 10 и 15?

Ссылка на видео-разбор с таймингом: [https://vk.com/video-205546952\\_456241277?t=0h3m15s](https://vk.com/video-205546952_456241277?t=0h3m15s)

### Решение

Напишем рекурсивную функцию, которая пройдет различные числовые траектории из одного числа до конечного и посчитает их количество, при этом будем учитывать, что траектория не должна содержать чисел 10 и 15. Функция будет принимать два параметра:

- с: текущее число.
- е: конечное число, к которому нужно прийти.

Рекурсивно она будет вычислять возможные пути от с до е.

Условия задачи:

1. Если текущее число с больше целевого числа е, или  $s == 10$ , или  $s == 15$ , то вернуть 0.
2. Если текущее число с равно целевому числу е, то вернуть 1.
3. Если текущее число с меньше целевого числа е, то вычислить возможные пути для трех действий:
  - Увеличить с на 1 и рекурсивно вызвать функцию.
  - Увеличить с на 2 и рекурсивно вызвать функцию.

• Увеличить  $c$  на 3 и рекурсивно вызвать функцию.

Выводим на экран результат умножения  $f(5,11)$  – числа возможных траекторий от 5 до 11 и  $f(11,18)$  – числа возможных траекторий от 11 до 18. Таким образом мы учитываем, что траектория вычислений содержит указанное в задаче число.

```
def f(c,e):  
    if c>e or c==10 or c==15: return 0  
    if c==e: return 1  
    if c<e: return f(c+1,e)+f(c+2,e)+f(c+3,e)
```

```
print(f(5,11)*f(11,18))
```

Ответ: 176

Telegram: @fast\_ege

## DZ\_23\_prog\_4

### Задача № 4 (1204)

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

1. Прибавить 1
2. Умножить на 2

Первая команда увеличивает число на экране на 1, вторая умножает его на 2. Программа для исполнителя Май — это последовательность команд.

Сколько существует программ, для которых при исходном числе 2 результатом является число 39 и при этом траектория вычислений содержит числа 7 и 16?

Траектория вычислений программы — это последовательность результатов выполнения всех команд программы. Например, для программы 121 при исходном числе 7 траектория будет состоять из чисел 8, 16, 17.

Ссылка на видео-разбор с таймингом: [https://vk.com/video-205546952\\_456241277?t=0h4m50s](https://vk.com/video-205546952_456241277?t=0h4m50s)

*Решение*

Напишем рекурсивную функцию, которая пройдет различные числовые траектории из одного числа до конечного и посчитает их количество. Функция будет принимать два параметра:

- $c$ : текущее число.
- $e$ : конечное число, к которому нужно прийти.

Рекурсивно она будет вычислять возможные пути от  $c$  до  $e$ .

Условия задачи:

1. Если текущее число  $c$  больше целевого числа  $e$ , то вернуть 0.
2. Если текущее число  $c$  равно целевому числу  $e$ , то вернуть 1.
3. Если текущее число  $c$  меньше целевого числа  $e$ , то вычислить возможные пути для двух действий:

- Увеличить  $c$  на 1 и рекурсивно вызвать функцию.
- Умножить  $c$  на 2 и рекурсивно вызвать функцию.

Выводим на экран результат умножения  $f(2,7)$  – числа возможных траекторий от 2 до 7,  $f(7,16)$  – числа возможных траекторий от 7 до 16 и  $f(16,39)$  – числа возможных траекторий от 16 до 39.

Таким образом мы учитываем, что траектория вычислений содержит указанные в задаче числа.

```
def f(c,e):  
    if c>e: return 0  
    if c==e: return 1  
    if c<e: return f(c+1,e)+f(c*2,e)
```

```
print(f(2,7)*f(7,16)*f(16,39))
```

Ответ: 45

## DZ\_23\_prog\_5

### Задача №5 (1301)

Исполнитель Минус преобразует число на экране.

У исполнителя есть две команды, которым присвоены номера:

1. Вычесть 2
2. Вычесть 5

Первая команда уменьшает число на экране на 2, вторая уменьшает это число на 5. Программа для исполнителя Минус – это последовательность команд. Сколько существует программ, которые число 23 преобразуют в число 2?

Ссылка на видео-разбор с таймингом: [https://vk.com/video-205546952\\_456241277?t=0h6m25s](https://vk.com/video-205546952_456241277?t=0h6m25s)

*Решение*

Напишем рекурсивную функцию, которая пройдет различные числовые траектории из одного числа до конечного и посчитает их количество. Функция будет принимать два параметра:

- с: текущее число.
- е: конечное число, к которому нужно прийти.

Рекурсивно она будет вычислять возможные пути от с до е.

Условия задачи:

1. Если текущее число с меньше целевого числа е, то вернуть 0.
2. Если текущее число с равно целевому числу е, то вернуть 1.
3. Если текущее число с больше целевого числа е, то вычислить возможные пути для двух действий:

- Уменьшить с на 2 и рекурсивно вызвать функцию.
- Уменьшить с на 5 и рекурсивно вызвать функцию.

Выводим на экран результат работы функции  $f(23, 2)$ , получаем ответ.

```
def f(c, e):  
    if c < e: return 0  
    if c == e: return 1  
    if c > e: return f(c-2, e) + f(c-5, e)
```

```
print(f(23, 2))
```

Ответ: 29

Telegram: @fast\_ege

## DZ\_23\_prog\_6

### Задание №6 (18047)

Исполнитель преобразует число на экране. У исполнителя есть три команды, которые обозначены латинскими буквами:

- A. Вычесть 2
- B. Вычесть 3
- C. Найти целую часть от деления на 4

Программа для исполнителя — это последовательность команд.

Сколько существует программ, для которых при исходном числе 36 результатом является число 13, при этом траектория вычислений не содержит числа 24?

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

Например, для программы СВА при исходном числе 51 траектория будет состоять из чисел 12, 9, 7.

Ссылка на видео-разбор с таймингом: [https://vk.com/video-205546952\\_456241277?t=0h7m35s](https://vk.com/video-205546952_456241277?t=0h7m35s)

*Решение*

Напишем рекурсивную функцию, которая пройдет различные числовые траектории из одного числа до конечного и посчитает их количество, при этом будем учитывать, что траектория не должна содержать число 24. Функция будет принимать два параметра:

- с: текущее число.
- е: конечное число, к которому нужно прийти.

Рекурсивно она будет вычислять возможные пути от с до е.

Условия задачи:

1. Если текущее число с меньше целевого числа е, или  $s == 24$ , то вернуть 0.
2. Если текущее число с равно целевому числу е, то вернуть 1.
3. Если текущее число с больше целевого числа е, то вычислить возможные пути для трех действий:
  - Уменьшить с на 2 и рекурсивно вызвать функцию.
  - Уменьшить с на 3 и рекурсивно вызвать функцию.
  - Найти целую часть от деления числа с на 4 и рекурсивно вызвать функцию.

Выводим на экран результат работы функции  $f(36,13)$  — числа возможных траекторий от 36 до 13.

```
def f(c,e):  
    if c<e or c==24: return 0  
    if c==e: return 1  
    if c>e: return f(c-2,e)+f(c-3,e)+f(c//4,e)  
  
print(f(36,13))
```

Ответ: 157

Telegram: @fast\_ege

## DZ\_23\_prog\_7

### Задание № 7(9844)

Исполнитель преобразует число на экране. У исполнителя есть три команды, которые обозначены латинскими буквами:

- А. Вычесть 1
- В. Вычесть 3
- С. Найти целую часть от деления на 2

Программа для исполнителя — это последовательность команд.

Сколько существует программ, для которых при исходном числе 19 результатом является число 3, при этом траектория вычислений не содержит числа 7 и содержит 10?

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

Например, для программы СВА при исходном числе 13 траектория будет состоять из чисел 6, 3, 2.

Ссылка на видео-разбор с таймингом: [https://vk.com/video-205546952\\_456241277?t=0h8m40s](https://vk.com/video-205546952_456241277?t=0h8m40s)

*Решение*

Напишем рекурсивную функцию, которая пройдет различные числовые траектории из одного числа до конечного и посчитает их количество, при этом будем учитывать, что траектория не должна содержать число 7. Функция будет принимать два параметра:

- с: текущее число.
- е: конечное число, к которому нужно прийти.

Рекурсивно она будет вычислять возможные пути от с до е.

Условия задачи:

1. Если текущее число  $c$  меньше целевого числа  $e$ , или  $c == 7$ , то вернуть 0.
2. Если текущее число  $c$  равно целевому числу  $e$ , то вернуть 1.
3. Если текущее число  $c$  больше целевого числа  $e$ , то вычислить возможные пути для трех действий:
  - Уменьшить  $c$  на 1 и рекурсивно вызвать функцию.
  - Уменьшить  $c$  на 3 и рекурсивно вызвать функцию.
  - Найти целую часть от деления числа  $c$  на 2 и рекурсивно вызвать функцию.

Выводим на экран результат умножения  $f(19,10)$  – числа возможных траекторий от 19 до 10 и  $f(10,3)$  – числа возможных траекторий от 10 до 3. Таким образом мы учитываем, что траектория вычислений содержит указанное в задаче число.

```
def f(c,e):  
    if c<e or c==7: return 0  
    if c==e: return 1  
    if c>e: return f(c-1,e)+f(c-3,e)+f(c//2,e)
```

```
print(f(19,10)*f(10,3))
```

Ответ: 133

Telegram: @fast\_\_ege

## DZ\_23\_prog\_8

### Задание №8 (18146)

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

- A. Вычти 3
- B. Найди целую часть от деления на 3
- C. Найди целую часть от деления на 2

Программа для исполнителя – это последовательность команд.

Сколько существует программ, для которых при исходном числе 46 результатом является число 3 и при этом траектория вычислений содержит число 20 и не содержит числа 28?

Траектория вычислений программы – это последовательность результатов выполнения всех команд программы. Например, для программы ABB при исходном числе 13 траектория состоит из чисел 11, 5, 2.

Ссылка на видео-разбор с таймингом: [https://vk.com/video-205546952\\_456241277?t=0h10m30s](https://vk.com/video-205546952_456241277?t=0h10m30s)

Решение

Напишем рекурсивную функцию, которая пройдет различные числовые траектории из одного числа до конечного и посчитает их количество, при этом будем учитывать, что траектория не должна содержать число 28. Функция будет принимать два параметра:

- $c$ : текущее число.
- $e$ : конечное число, к которому нужно прийти.

Рекурсивно она будет вычислять возможные пути от  $c$  до  $e$ .

Условия задачи:

1. Если текущее число  $c$  меньше целевого числа  $e$ , или  $c == 28$ , то вернуть 0.
2. Если текущее число  $c$  равно целевому числу  $e$ , то вернуть 1.
3. Если текущее число  $c$  больше целевого числа  $e$ , то вычислить возможные пути для трех действий:
  - Уменьшить  $c$  на 1 и рекурсивно вызвать функцию.
  - Найти целую часть от деления числа  $c$  на 3 и рекурсивно вызвать функцию.
  - Найти целую часть от деления числа  $c$  на 2 и рекурсивно вызвать функцию.

Выводим на экран результат умножения  $f(46,20)$  – числа возможных траекторий от 46 до 20 и  $f(20,3)$  – числа возможных траекторий от 11 до 18. Таким образом мы учитываем, что траектория

вычислений содержит указанное в задаче число.

```
def f(c,e):  
    if c<e or c==28: return 0  
    if c==e: return 1  
    if c>e: return f(c-3,e)+f(c//3,e)+f(c//2,e)  
  
print(f(46,20)*f(20,3))
```

Ответ: 12

Telegram: @fast\_ege

## DZ\_23\_prog\_9

### Задание №9(1760)

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

1. Прибавить 2
2. Прибавить 3
3. Прибавить 5

Сколько существует программ, для которых при исходном числе 5 результатом является число 25, и при этом траектория вычислений содержит либо 13, либо 17, но не 13 и 17 одновременно.

Ссылка на видео-разбор с таймингом: [https://vk.com/video-205546952\\_456241277?t=0h11m55s](https://vk.com/video-205546952_456241277?t=0h11m55s)

*Решение*

Напишем две рекурсивные функции, которые пройдут различные числовые траектории из одного числа до конечного и посчитают их количество, при этом будем учитывать, что траектория не должна содержать числа 13 в одном случае и числа 17 в другом случае. Функция будет принимать два параметра:

- с: текущее число.
- е: конечное число, к которому нужно прийти.

Рекурсивно она будет вычислять возможные пути от с до е.

Первый случай:

1. Если текущее число с больше целевого числа е, или  $s == 17$ , то вернуть 0.
2. Если текущее число с равно целевому числу е, то вернуть 1.
3. Если текущее число с меньше целевого числа е, то вычислить возможные пути для трех действий:
  - Увеличить с на 2 и рекурсивно вызвать функцию.
  - Увеличить с на 3 и рекурсивно вызвать функцию.
  - Увеличить с на 5 и рекурсивно вызвать функцию.

Выводим на экран результат умножения  $f(5,13)$  – числа возможных траекторий от 5 до 13 и  $f(13,25)$  – числа возможных траекторий от 13 до 25. Таким образом мы учитываем, что траектория вычислений содержит число 13. Получили 126 возможных траекторий.

Второй случай:

1. Если текущее число с больше целевого числа е, или  $s == 13$ , то вернуть 0.
2. Если текущее число с равно целевому числу е, то вернуть 1.
3. Если текущее число с меньше целевого числа е, то вычислить возможные пути для трех действий:
  - Увеличить с на 2 и рекурсивно вызвать функцию.
  - Увеличить с на 3 и рекурсивно вызвать функцию.
  - Увеличить с на 5 и рекурсивно вызвать функцию.

Выводим на экран результат умножения  $f(5,17)$  – числа возможных траекторий от 5 до 17 и  $f(17,25)$  – числа возможных траекторий от 17 до 25. Таким образом мы учитываем, что траектория вычислений содержит число 17. Получили 126 возможных траекторий.

Для получения ответа на задачу, надо сложить количество траекторий, полученной в первом случае, с количеством траекторий, полученном во втором.  $126 + 126 = 252$ .

```
def f(c,e):
    if c>e or c==17: return 0
    if c==e: return 1
    if c<e: return f(c+2,e)+f(c+3,e)+f(c+5,e)

print(f(5,13)*f(13,25))
```

```
def f(c,e):
    if c>e or c==13: return 0
    if c==e: return 1
    if c<e: return f(c+2,e)+f(c+3,e)+f(c+5,e)

print(f(5,17)*f(17,25))
```

Ответ: 252

Telegram: @fast\_ege

## DZ\_23\_prog\_10

### Задание №10 (2340)

У исполнителя Калькулятор есть три команды, которым присвоены номера:

1. Прибавить 2
2. Прибавить 4
3. Прибавить 5

Определите число, для получения которого из числа 31 существует 1001 программа.

Ссылка на видео-разбор с таймингом: [https://vk.com/video-205546952\\_456241277?t=0h14m50s](https://vk.com/video-205546952_456241277?t=0h14m50s)

*Решение*

Напишем рекурсивную функцию, которая пройдет различные числовые траектории из одного числа до конечного и посчитает их количество. Функция будет принимать два параметра:

- с: текущее число.
- е: конечное число, к которому нужно прийти.

Рекурсивно она будет вычислять возможные пути от с до е.

Условия задачи:

1. Если текущее число с больше целевого числа е, то вернуть 0.
2. Если текущее число с равно целевому числу е, то вернуть 1.
3. Если текущее число с меньше целевого числа е, то вычислить возможные пути для трех действий:
  - Увеличить с на 2 и рекурсивно вызвать функцию.
  - Увеличить с на 4 и рекурсивно вызвать функцию.
  - Увеличить с на 5 и рекурсивно вызвать функцию.

Будем перебирать числа е от 32 до 100, и если функция  $f(31, e)$  будет равна 1001, выведем число е на экран.

```
def f(c,e):
    if c>e: return 0
    if c==e: return 1
    if c<e: return f(c+2,e)+f(c+4,e)+f(c+5,e)

for e in range(32,100):
    if f(31,e)==1001:
        print(e)
```



Ответ: 56

Telegram: @fast\_ege

## DZ\_23\_prog\_11

### Задача № 11(7687)

Исполнитель Великий Отрицатель преобразует число, записанное на доске. У Великого Отрицателя есть три команды:

1. Вычесть 3
2. Разделить на 3
3. Отбросить последнюю цифру

Первая команда уменьшает число на 3, вторая команда делит его без остатка на 3, третья убирает последнюю цифру из числа.

Команда 2 применяется только для четных чисел.

Команда 3 применяется к числам, состоящим хотя бы из 2-х цифр.

Сколько существует программ, для которых при исходном числе 1250 результатом является число 20?

Ссылка на видео-разбор с таймингом: [https://vk.com/video-205546952\\_456241277?t=0h16m50s](https://vk.com/video-205546952_456241277?t=0h16m50s)

#### Решение

Напишем рекурсивную функцию, которая пройдет различные числовые траектории из одного числа до конечного и посчитает их количество. Функция будет принимать два параметра:

- с: текущее число.
- е: конечное число, к которому нужно прийти.

Рекурсивно она будет вычислять возможные пути от с до е.

Условия задачи:

1. Если текущее число с меньше целевого числа е, то вернуть 0.
2. Если текущее число с равно целевому числу е, то вернуть 1.
3. Если текущее число с больше целевого числа е, то напишем сложную обработку: Пусть k – количество траекторий:

•  $k = f(c - 3, e)$ .

• Если с четное число, то добавляем к k количество траекторий  $f(c//3, e)$ , результат выполнения команды 2.

• Если с больше или равно 10 (т.е. как минимум двузначное число), то добавляем к k количество траекторий  $f(c//10, e)$ , результат выполнения команды 3.

Возвращаем результат k.

Выводим на экран значение функции  $f(1250, 20)$ , это число будет ответом на задачу.

```
def f(c,e):
    if c<e: return 0
    if c==e: return 1
    if c>e:
        k = f(c-3,e)
        if c%2==0: k+= f(c//3,e)
        if c>=10: k += f(c//10,e)
        return k
print(f(1250,20))
```

Ответ: 8492

Telegram: @fast\_ege

## DZ\_23\_prog\_12

### Задача № 12 (14418)

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

1. Прибавить  $d$
2. Умножить на 2

Первая команда увеличивает число на экране на  $d$  ( $d$  - натуральное число), вторая умножает его на

2. Программа для исполнителя – это последовательность команд.

Посчитали количество различных программ, для которых при исходном числе 1 результатом является число 100, и при этом траектория вычислений содержит число 10. Укажите число  $d$ , при котором количество таких программ будет равно 13.

Траектория вычислений программы – это последовательность результатов выполнения всех команд программы. Например, для программы 121 при исходном числе 7 и  $d = 1$  траектория будет состоять из чисел 8, 16, 17.

Ссылка на видео-разбор с таймингом: [https://vk.com/video-205546952\\_456241277?t=0h19m25s](https://vk.com/video-205546952_456241277?t=0h19m25s)

*Решение*

Напишем рекурсивную функцию, которая пройдет различные числовые траектории из одного числа до конечного и посчитает их количество. Функция будет принимать три параметра:

- $c$ : текущее число.
- $e$ : конечное число, к которому нужно прийти.
- $d$ : некоторое число, на которое можно увеличить  $c$

Рекурсивно она будет вычислять возможные пути от  $c$  до  $e$ .

Условия задачи:

1. Если текущее число  $c$  больше целевого числа  $e$ , то вернуть 0.
2. Если текущее число  $c$  равно целевому числу  $e$ , то вернуть 1.
3. Если текущее число  $c$  меньше целевого числа  $e$ , то вычислить возможные пути для двух действий:
  - Увеличить  $c$  на  $d$  и рекурсивно вызвать функцию.
  - Умножить  $c$  на 2 и рекурсивно вызвать функцию.

Будем перебирать числа  $d$  от 1 до 100, и если результат перемножения значений функции  $f(1, 10, d)$  и функции  $f(10, 100, d)$  будет равен 13, выведем число  $d$  на экран.

```
def f(c, e, d):  
    if c > e: return 0  
    if c == e: return 1  
    if c < e: return f(c + d, e, d) + f(c * 2, e, d)  
  
for d in range(1, 100):  
    if f(1, 10, d) * f(10, 100, d) == 13:  
        print(d)
```

Ответ: 6

Telegram: @fast\_ege