

## . Текстовый разбор домашки 1

### DZ\_5\_1

#### Задача № 1 (548)

На вход алгоритма подаётся натуральное число  $N$ . Алгоритм строит по нему новое число  $R$  следующим образом.

- 1) Строится двоичная запись числа  $N$ .
- 2) К этой записи дописывается (дублируется) последняя цифра.
- 3) Затем справа дописывается бит чётности: 0, если в двоичном коде полученного числа чётное число единиц, и 1, если нечётное.
- 4) К полученному результату дописывается ещё один бит чётности.

Полученная таким образом запись (в ней на три разряда больше, чем в записи исходного числа  $N$ ) является двоичной записью искомого числа  $R$ . Укажите минимальное число  $R$ , большее 114, которое может быть получено в результате работы этого алгоритма. В ответе это число запишите в десятичной системе.

Ссылка на видео-разбор с таймингом: [https://vk.com/video-205546952\\_456241250?t=0h0m0s](https://vk.com/video-205546952_456241250?t=0h0m0s)

#### Решение

Для решения этой задачи переберем значения  $n$  в диапазоне от 1 до 500. Для каждого числа ( $n$ ) выполним следующие шаги:

1. Создадим строковое двоичное представление числа  $n$  с использованием f-строки `f'{n:b}'`.
2. Продублируем последнюю цифру числа.
3. Проверим четность количества единиц в полученной строке. Если их количество четное, добавим 0, иначе — 1. Эта операция выполняется дважды.
4. Полученную строку преобразуем обратно в десятичное число с помощью функции `int` и проверим условие  $r > 114$ .
5. Все подходящие значения сохраним в списке `m`.
6. В результате выводится минимальное значение из списка `m`.

```
m = []
for n in range(1, 500):
    b = f'{n:b}'
    b = b + b[-1]
    if b.count('1') % 2 == 0:
        b = b + '0'
    else:
        b = b + '1'
    if b.count('1') % 2 == 0:
        b = b + '0'
    else:
        b = b + '1'
    r = int(b, 2)
```

```
if r>114:
    m.append(r)
print(min(m))
```

Результат работы программы:

126

Ответ: 126

Telegram: @fast\_ege

## DZ\_5\_2

### Задача №2 (1511)

На вход алгоритма подаётся натуральное число  $N$ . Алгоритм строит по нему новое число  $R$  следующим образом.

- 1) Строится двоичная запись числа  $N$ .
- 2) К этой записи дописывается справа бит чётности: 0, если в двоичном коде числа  $N$  было чётное число единиц, и 1, если нечётное.
- 3) К полученному результату дописывается ещё один бит чётности.

Полученная таким образом запись (в ней на два разряда больше, чем в записи исходного числа  $N$ ) является двоичной записью искомого числа  $R$ . Укажите количество исходных чисел  $N$ , из которых с помощью этого алгоритма могут быть получены числа  $R$ , лежащие в диапазоне  $64 \leq R < 72$ .

Ссылка на видео-разбор с таймингом: [https://vk.com/video-205546952\\_456241250?t=0h2m50s](https://vk.com/video-205546952_456241250?t=0h2m50s)

#### Решение

Для решения задачи создадим список и будем перебирать различные значения  $n$  в цикле `for`. Ограничимся значениями от 1 до 500, так как этого диапазона будет достаточно.

Создадим строковое двоичное представление числа  $n$  с использованием `f`-строки `f'{n:b}'`. Если количество единиц в полученной строке чётно, добавляем '0', иначе добавляем '1'. Повторим эту операцию ещё раз для нового результата. Полученное число переведём в десятичное представление и проверим, попадает ли оно в диапазон от 64 до 72. Если да, добавим соответствующее значение  $n$  в наш список.

В завершение выведем количество элементов в списке, которое соответствует количеству подходящих значений  $n$ .

```
m = []
for n in range(1, 500):
    b = f'{n:b}'
    if b.count('1') % 2 == 0:
        b = b + '0'
    else:
        b = b + '1'
    if b.count('1') % 2 == 0:
        b = b + '0'
    else:
        b = b + '1'
    r = int(b, 2)
    if 64 <= r < 72:
        m.append(n)
print(len(m))
```

Результат работы программы:

2  
Запустив программу, получаем длину списка равную двум.  
Ответ: 2  
Telegram: @fast\_ege

## DZ\_5\_3

### Задача № 3 (1188)

На вход алгоритма подаётся натуральное число  $N$ . Алгоритм строит по нему новое число  $R$  следующим образом.

1. Строится двоичная запись числа  $N$ .

2. К этой записи дописываются ещё два разряда по следующему правилу:

- а) если  $N$  чётное, то к нему справа приписываются две последние цифры его двоичной записи;
- б) если  $N$  нечётное, то к нему справа и слева приписывается цифра 1.

Полученная таким образом запись (в ней на два разряда больше, чем в записи исходного числа  $N$ ) является двоичной записью искомого числа  $R$ .

Например, двоичная запись нечётного числа  $11001_2$  будет преобразована в  $1110011_2$ .

Укажите такое наименьшее число  $R$ , превышающее 130, которое может являться результатом работы данного алгоритма.

В ответе это число запишите в десятичной системе счисления.

Ссылка на видео-разбор с таймингом: [https://vk.com/video-205546952\\_456241250?t=0h4m45s](https://vk.com/video-205546952_456241250?t=0h4m45s)

*Решение*

Создадим строковое двоичное представление числа  $n$  с использованием f-строки `f'{n:b}'`. Если  $n$  чётное, то к этому числу добавляются две последние цифры его двоичного представления. Если  $n$  нечётное, то добавляется единица как спереди, так и сзади.

Нужно найти минимальное значение  $r$ , большее 130, которое может быть получено в результате выполнения описанного алгоритма. Для поиска такого числа перебираются различные значения  $n$  от 1 до 300, этого диапазона должно хватить.

Если  $n$  чётно, то к числу добавляются две последние цифры его двоичной записи. В противном случае, добавляется единица с обеих сторон. После преобразования результата обратно в десятичное число проверяется условие: если оно превышает 130, то записывается в список  $m$ . Наконец, выводится минимальное значение из списка  $m$ .

```
m = []
for n in range(1, 300):
    b = f'{n:b}'
    if n%2==0:
        b = b + b[-2]+b[-1]
    else:
        b = '1' + b + '1'
    r = int(b, 2)
    if r>130:
        m.append(r)
print(min(m))
```

Результат работы программы:

138

Результат вычислений показывает, что минимальным числом, удовлетворяющим условию задачи, является 138.

Ответ: 138

Telegram: @fast\_ege

## DZ\_5\_4

### Задача № 4 (2231)

На вход алгоритма подаётся натуральное число  $N$ . Алгоритм строит по нему новое число  $R$  следующим образом:

- 1) Строится двоичная запись числа  $N$ .
- 2) К этой записи дописывается ещё три или четыре разряда по следующему правилу: если  $N$  нечётное, то слева к нему приписывается "10", а справа - "11". В противном случае слева приписывается "1", а справа "00".

Например,  $N = 5_{10} = 101_2 \Rightarrow 1010111_2 = 87_{10} = R$

Полученная таким образом запись (в ней на три или четыре разряда больше, чем в записи исходного числа  $N$ ) является двоичной записью искомого числа  $R$ . Укажите наименьшее число  $R$ , большее 1023, которое может быть получено с помощью описанного алгоритма. В ответ запишите это число в десятичной системе счисления.

Ссылка на видео-разбор с таймингом: [https://vk.com/video-205546952\\_456241250?t=0h6m25s](https://vk.com/video-205546952_456241250?t=0h6m25s)

### Решение

Для решения задачи создадим список  $m$  и будем последовательно перебирать различные значения  $n$ , например, от 1 до 1000. Для каждого  $n$  строим его двоичное представление с использованием f-строки `f'{n:b}'`. Если  $n$  чётное, добавляем слева "1" и справа "00"; если  $n$  нечётное, добавляем слева "10" и справа "11".

После преобразования строки в десятичное число проверяем, превышает ли оно 1023. Если условие выполняется, добавляем это число в список  $m$ . Наконец, находим минимальное значение среди всех чисел в списке  $m$ .

```
m = []
for n in range(1, 1000):
    b = f'{n:b}'
    if n%2!=0:
        b = '10'+b+'11'
    else:
        b = '1'+b+'00'
    r = int(b, 2)
    if r>1023:
        m.append(r)
print(min(m))
```

Результат работы программы:

1287

Ответ:1287

Telegram: @fast\_ege

## DZ\_5\_5

### Задача №5 (350)

На вход алгоритма подаётся натуральное число  $N$ . Алгоритм строит по нему новое число  $R$  следующим образом.

- 1) Строится двоичная запись числа  $N$ .
- 2) К этой записи дописываются разряды по следующему правилу:
  - а) если единиц больше, чем нулей, в конец дописывается 0,
  - б) иначе в начало числа дописывается 11.
- 3) Повторяется пункт 2

Полученная таким образом запись является двоичной записью искомого числа  $R$ .

Укажите минимальное число  $N$ , при вводе которого получится значение  $R$  больше, чем 500. В ответе полученное число запишите в десятичной системе.

Ссылка на видео-разбор с таймингом: [https://vk.com/video-205546952\\_456241250?t=0h8m35s](https://vk.com/video-205546952_456241250?t=0h8m35s)

### Решение

Для решения задачи будем последовательно проверять различные значения  $n$  в диапазоне от 1 до 500.

Для каждого  $n$  строим его двоичное представление с использованием f-строки `f'{n:b}'`. Если количество единиц превышает количество нулей, добавляем 0 в конец строки. В противном случае, добавляем две единицы в начало строки. Повторяем эту операцию ещё раз. Далее преобразуем полученную строку обратно в десятичное число. Если оно больше 500, сохраняем соответствующее значение  $n$ .

```
m = []
for n in range(1, 500):
    b = f'{n:b}'
    if b.count('1') > b.count('0'):
        b = b + '0'
    else:
        b = '11' + b
    if b.count('1') > b.count('0'):
        b = b + '0'
    else:
        b = '11' + b
    r = int(b, 2)
    if r > 500:
        m.append(n)
print(min(m))
```

По завершении проверки всех значений  $n$ , находим минимальное среди тех, которые удовлетворяют условию.

Результат работы программы :

32

Ответ: 32

Telegram: @fast\_ege

## DZ\_5\_6

### Задание №6 (1116)

Алгоритм получает на вход натуральное число  $N > 1$  и строит по нему новое число  $R$  следующим образом:

- 1) Строится двоичная запись числа  $N$ .
- 2) Подсчитывается количество нулей и единиц в полученной записи. Если их количество одинаково, в конец записи добавляется её последняя цифра. В противном случае в конец записи добавляется цифра, которая встречается реже.
- 3) Шаг 2 повторяется ещё два раза.
- 4) Результат переводится в десятичную систему счисления.

При каком наибольшем исходном числе  $N < 100$  в результате работы алгоритма получится число, которое делится на 4 и не делится на 8?

Ссылка на видео-разбор с таймингом: [https://vk.com/video-205546952\\_456241250?t=0h10m30s](https://vk.com/video-205546952_456241250?t=0h10m30s)

### Решение

Для решения данной задачи необходимо создать пустой список  $m$ , который будет использоваться для хранения значений переменной  $n$ . Затем нужно перебрать все целые числа  $n$  от 1 до 99 включительно, поскольку по условию задачи  $n < 100$ .

Для каждого значения  $n$  строится его двоичная запись с использованием f-строки  $f'\{n:b\}'$ . После этого проводится анализ этой записи:

1. Если количество нулей и единиц в двоичной записи одинаково, то последняя цифра двоичного представления добавляется к концу строки.
2. Если количество нулей больше количества единиц, то к концу строки добавляется единица '1'.
3. В противном случае (если единиц больше) к концу строки добавляется ноль '0'.

Этот процесс преобразования двоичной записи повторяется три раза.

После завершения преобразований полученная строка переводится обратно в десятичную систему счисления. Полученное число проверяется на два условия:

- Оно должно быть кратно 4.
- Оно не должно быть кратно 8.

Если оба этих условия выполнены, соответствующее значение  $n$  сохраняется в списке  $m$ .

По завершении цикла программа находит максимальное значение среди элементов списка  $m$  и выводит его.

```
m = []
for n in range(1,100):
    b = f'{n:b}'
    for i in range(3):
        if b.count('0')==b.count('1'):
            b = b + b[-1]
        elif b.count('0')>b.count('1'):
            b = b + '1'
        else:
            b = b + '0'
    r = int(b,2)
    if r%4==0 and r%8!=0:
        m.append(n)
print(max(m))
```

Результат работы программы:

49

Обратите внимание на использование операторов ветвления if, elif и else. Они играют важную роль в корректной реализации алгоритма. Условия проверяются последовательно:

1. Сначала проверяется условие в блоке if: если количество нулей равно количеству единиц, то к строке добавляется последняя цифра.
2. Если это условие не выполняется, проверяется второе условие elif: если количество нулей больше количества единиц, то к строке добавляется '1'.

3. Если ни одно из предыдущих условий не выполнено, срабатывает блок `else`, где к строке добавляется '0', так как в этом случае единиц больше, чем нулей.

Ответ: 49

Telegram: @fast\_ege

## DZ\_5\_7

### Задание № 7(4665)

На вход алгоритма подаётся натуральное число  $N$ . Алгоритм строит по нему новое число  $R$  следующим образом.

1. Строится двоичная запись числа  $N$ .
2. Далее эта запись обрабатывается по следующему правилу:
  - а) если сумма цифр в двоичной записи числа чётная, то к этой записи справа дописывается 1, а затем два левых разряда заменяются на 10;
  - б) если сумма цифр в двоичной записи числа нечётная, то к этой записи справа дописывается 0, а затем два левых разряда заменяются на 11.

Полученная таким образом запись является двоичной записью искомого числа  $R$ .

Например, для исходного числа  $6_{10} = 110_2$  результатом является число  $1001_2 = 9_{10}$ , а для исходного числа  $4_{10} = 100_2$  результатом является число  $1100_2 = 12_{10}$ .

Укажите максимальное число  $R$ , которое может быть получено при обработке числа  $N$ , меньшего 16. В ответе запишите это число в десятичной системе счисления.

Ссылка на видео-разбор с таймингом: [https://vk.com/video-205546952\\_456241250?t=0h13m40s](https://vk.com/video-205546952_456241250?t=0h13m40s)

### Решение

Для решения задачи необходимо последовательно обработать все числа  $n$  в диапазоне от 1 до 14 включительно. Для каждого числа выполняются следующие шаги:

1. Строится двоичная запись числа  $n$ .
2. Определяется сумма цифр этой записи, что эквивалентно подсчёту количества единиц в ней.
3. Применяется соответствующее правило обработки записи в зависимости от чётности суммы цифр.
4. Полученная строка преобразуется обратно в десятичную систему счисления, и результат сохраняется в массиве  $m$ .

По завершении обработки всех чисел определяется максимальное значение среди элементов массива  $m$ .

```
m = []
for n in range(1,15):
    b = f'{n:b}'
    if b.count('1')%2==0:
        b = b + '1'
        b = '10' + b[2:]
    else:
        b = b + '0'
```



```
b = '11' + b[2:]
r = int(b, 2)
m.append(r)
print(max(m))
```

Результат работы программы:

30

Ответ: 30

Telegram: @fast\_ege

## DZ\_5\_8

### Задание №8 (6012)

На вход алгоритма подаётся натуральное число  $N$ . Алгоритм строит по нему новое число  $R$  следующим образом.

1. Строится двоичная запись числа  $N$ .
2. Далее эта запись обрабатывается по следующему правилу:
  - а) если сумма цифр в двоичной записи числа чётная, то к этой записи справа дописывается 0, а затем два левых разряда заменяются на 1;
  - б) если сумма цифр в двоичной записи числа нечётная, то к этой записи справа дописывается 1, а затем два левых разряда заменяются на 11.

Полученная таким образом запись является двоичной записью искомого числа  $R$ .

Например, для исходного числа  $6_{10} = 110_2$  результатом является число  $100_2 = 4_{10}$ , а для исходного числа  $4_{10} = 100_2$  результатом является число  $1101_2 = 13_{10}$ .

Укажите число  $N$ , после обработки которого с помощью этого алгоритма получается наименьшее значение  $R$ , большее 49. В ответе запишите это число в десятичной системе счисления.

Ссылка на видео-разбор с таймингом: [https://vk.com/video-205546952\\_456241250?t=0h16m25s](https://vk.com/video-205546952_456241250?t=0h16m25s)

### Решение

Важно отметить, что необходимо вывести исходное число  $n$ , а не итоговое значение  $r$ .

Рассмотрим пошагово решение данной задачи.

1. Последовательно перебираем все числа  $n$  от 1 до 100. Для каждого числа находим его двоичное представление.
2. Проверяем, является ли количество единиц в полученной двоичной записи четным числом. Если да, добавляем к числу 0 и устанавливаем два старших разряда в 1. Иначе добавляем 1 и также изменяем два старших разряда на 1.
3. Переводим получившееся двоичное число обратно в десятичный формат. Если полученный результат  $r$  больше 49, сохраняем пару  $(r, n)$  в список.
4. По окончании цикла выводим минимальную пару, где первый элемент соответствует минимальному  $r$ , большему 49, а второй элемент — исходному числу  $n$ .

```
m = []
for n in range(1, 100):
    b = f'{n:b}'
    if b.count('1') % 2 == 0:
```

```

        b = b + '0'
        b = '1' + b[2:]
    else:
        b = b + '1'
        b = '11' + b[2:]
    r = int(b, 2)
    if r > 49:
        m.append([r, n])
print(min(m))

```

Результат работы программы:

[50, 57]

В итоге минимальный результат  $r$ , который можно получить, равен 50, и он достигается при  $n = 57$ .

Ответ: 57

Telegram: @fast\_ege

## DZ\_5\_9

### Задание №9(18861)

На вход алгоритма подаётся натуральное число  $N > 3$ . Алгоритм строит по нему новое число  $R$  следующим образом.

1. Строится троичная запись числа  $N$ .
2. Далее эта запись обрабатывается по следующему правилу:  
если число оканчивается на 10, то к троичной записи числа слева дописывается 2, иначе дописывается 1.

Полученная таким образом запись является троичной записью искомого числа.

3. Результат переводится в десятичную систему и выводится на экран.

Например, исходное число  $1010 = 101_3$ , оканчивается на 01, значит преобразуется в число  $1101_3$ , результат в десятичной системе -  $37_{10}$ . Укажите минимальное натуральное число  $N$ , при котором результат работы данного алгоритма будет больше 130. В ответе запишите это число в десятичной системе счисления.

Ссылка на видео-разбор с таймингом: [https://vk.com/video-205546952\\_456241250?t=0h19m50s](https://vk.com/video-205546952_456241250?t=0h19m50s)

*Решение*

Для начала необходимо создать функцию, которая будет возвращать строку, представляющую собой троичное представление числа. Внутри этой функции создается пустая строка, а затем с помощью цикла `while` к ней добавляются остатки от деления числа на три, пока само число не станет равно нулю.

Когда мы получаем троичное представление числа `n`, проверяется последняя пара цифр этого числа. Если эти цифры составляют последовательность 10, то слева к числу добавляется 2. В противном случае добавляется 1. После этого полученное число переводится обратно в десятичное представление и сравнивается с числом 130. Если результат оказывается больше 130, то исходное число `n` добавляется в список.

По завершении работы программы находится минимальное число `n` среди всех чисел, которые были сохранены в списке, и выводится на экран.

```
def c3(x):
    s = ''
    while x>0:
        s = str(x%3) + s
        x = x//3
    return s

m = []
for n in range(4,200):
    b = c3(n)
    if b[-2]+b[-1]=='10':
        b = '2' + b
    else:
        b = '1' + b
    r = int(b,3)
    if r>130:
        m.append(n)
print(min(m))
```

Результат работы программы:

30

Ответ:30

Telegram: @fast\_ege

## DZ\_5\_10

### Задание №10 (10657)

На вход алгоритма подаётся натуральное число `N`. Алгоритм строит по нему новое число `R` следующим образом:

1. Строится троичная запись числа  $N$ .
  2. К этой записи дописываются разряды по следующему правилу. Если сумма троичных разрядов кратна 3, слева дописывается 20, иначе 10.
  3. Полученная таким образом запись является троичной записью искомого числа  $R$ .
- Например, для числа 10 троичная запись  $101_3$  преобразуется в запись  $10101_3 = 91$ , для числа 11 троичная запись  $102_3$  преобразуется в  $20102_3 = 173$ .
- Укажите максимальное значение  $N$ , после обработки которого с помощью этого алгоритма получается число  $R$ , меньшее чем 100.

Ссылка на видео-разбор с таймингом: [https://vk.com/video-205546952\\_456241250?t=0h22m15s](https://vk.com/video-205546952_456241250?t=0h22m15s)

### Решение

Для начала определим функцию, которая будет преобразовывать число в его троичное представление. Далее создадим список для хранения значений  $n$ , удовлетворяющих условию задачи. Для каждого значения  $n$  будем вычислять его троичное представление и проверять, делится ли сумма его цифр на 3. В зависимости от результата проверки добавляем либо 20, либо 10 перед этим представлением. Затем переведем полученную строку обратно в десятичное число и проверим, выполняется ли условие  $r < 100$ .

Если данное условие выполнено, сохраняем текущее значение  $n$  в списке. По завершении цикла выведем максимальное значение из списка, которое и будет искомым результатом.

```
def c3(x):
    s = ''
    while x>0:
        s = str(x%3)+s
        x = x//3
    return s

m = []
for n in range(1,100):
    b = c3(n)
    if sum(int(d) for d in b)%3==0:
        b = '20'+b
    else:
        b = '10'+b
    r = int(b,3)
    if r<100:
        m.append(n)
print(max(m))
```

Результат работы программы:

18

Ответ : 18

Telegram: @fast\_ege

## Задача № 11(14329)

На вход алгоритма подаётся натуральное число  $N$ . Алгоритм строит по нему новое число  $R$  следующим образом.

1. Строится четверичная запись числа  $N$ .
2. Далее эта запись обрабатывается по следующему правилу: а) если число  $N$  делится на 4, то к этой записи дописываются две первые четверичные цифры; б) если число  $N$  на 4 не делится, то остаток от деления умножается на 4, переводится в четверичную запись и дописывается в конец числа. Полученная таким образом запись является четверичной записью искомого числа  $R$ .
3. Результат переводится в десятичную систему и выводится на экран.

Например, для исходного числа  $11 = 23_4$  результатом является число  $2330_4 = 188$ , а для исходного числа  $12 = 30_4$  это число  $3030_4 = 204$ . Укажите минимальное число  $R$ , большее 291, которое может быть получено с помощью описанного алгоритма. В ответе запишите это число в десятичной системе счисления.

Ссылка на видео-разбор с таймингом: [https://vk.com/video-205546952\\_456241250?t=0h24m25s](https://vk.com/video-205546952_456241250?t=0h24m25s)

### Решение

Для начала создадим функцию для преобразования числа в четверичную систему счисления. Она будет работать следующим образом: пока число  $x$  больше 0, оно преобразуется в строку путем добавления остатка от деления на 4 к уже существующей строке. Затем число  $x$  делится нацело на 4.

Далее создадим список  $m$  для хранения результатов вычислений. Для каждого числа  $n$  в диапазоне от 1 до 299 включительно выполняем следующие действия:

1. Преобразуем число  $n$  в четверичную систему счисления.
2. Проверяем, делится ли число  $n$  на 4. Если да, добавляем к его четверичной записи две первых цифры.
3. Если нет, остаток от деления числа  $n$  на 4 умножаем на 4, преобразуем результат в четверичную систему и добавляем к числу.
4. Переводим полученную запись обратно в десятичное представление и проверяем, превышает ли она 291. Если да, сохраняем её в списке  $m$ .

После завершения цикла выводим минимальное значение из списка  $m$

```

s = ''
while x>0:
    s = str(x%4)+s
    x = x//4
return s

m = []
for n in range(1,300):
    b = c4(n)
    if n%4==0:
        b = b + b[0]+b[1]
    else:
        b = b + c4(n%4*4)
    r = int(b,4)
    if r>291:
        m.append(r)
print(min(m))

```

Результат работы программы:

296

Ответ: 296

Telegram: @fast\_\_ege

## DZ\_5\_12

### Задача № 12(14328)

На вход алгоритма подаётся натуральное число  $N$ . Алгоритм строит по нему новое число  $R$  следующим образом.

1. Строится запись числа  $N$  в системе счисления с основанием 12.
  2. Далее эта запись обрабатывается по следующему правилу:
    - а) если число  $N$  делится на 3, то слева к нему приписывается «1», а справа «B»;
    - б) если число  $N$  на 3 не делится, то слева к нему приписывается «2», а справа «0». Полученная таким образом запись является двенадцатеричной записью искомого числа  $R$ .
  3. Результат переводится в десятичную систему и выводится на экран.
- Например, для исходного числа  $11 = B_{12}$  результатом является число  $2B0_{12} = 420$ , а для исходного числа  $12 = 10_{12}$  это число  $110B_{12} = 1883$ .

Укажите максимальное число  $R$ , меньшее 1996, которое может быть получено с помощью описанного алгоритма. В ответе запишите это число в десятичной

системе счисления.

Ссылка на видео-разбор с таймингом: [https://vk.com/video-205546952\\_456241250?t=0h27m25s](https://vk.com/video-205546952_456241250?t=0h27m25s)

### Решение

Для решения этой задачи определим функцию, которая будет осуществлять перевод числа из десятичной системы счисления в систему с основанием 12. Для этого подключим модуль `string`, который предоставляет удобный способ работы со строками символов, и воспользуемся функцией `printable`.

Функция перевода в систему счисления с основанием 12 будет выглядеть следующим образом:

```
from string import printable
def c12(x):
    s = ''
    while x>0:
        s = printable[x%12] + s
        x = x//12
    return s
```

После создания функции перевода в систему счисления с основанием 12, используем её для преобразования чисел в диапазоне от 1 до 2000. Для каждого числа проверяем условие деления на 3 и выполняем соответствующие действия согласно заданному алгоритму. Затем преобразованное число переводим обратно в десятичное представление и сохраняем те значения, которые не превышают 1196.

```
from string import printable
def c12(x):
    s = ''
    while x>0:
        s = printable[x%12] + s
        x = x//12
    return s
m = []
for n in range(1,2000):
    b = c12(n)
    if n%3==0:
        b = '1' + b + 'b'
    else:
```

```
b = '2' + b + '0'
r = int(b, 12)
if r < 1996:
    m.append(r)
print(max(m))
```

Результат работы программы:

1991

Алгоритм нашёл максимальное значение  $r$ , меньшее 1196.

Ответ: 1991

Telegram: @fast\_ege

## DZ\_5\_13

### Задача № 13(5412)

На вход алгоритма подаётся натуральное число  $N$ . Алгоритм строит по нему новое число  $R$  следующим образом:

1. Строится шестнадцатеричная запись числа.
2. Далее, эта запись обрабатывается по следующему правилу:
  - а) Если цифр  $B$  в ней чётное количество, то к этой записи слева дописывается 1.
  - б) Если цифр  $B$  в ней нечётное количество, то к этой записи справа дописывается 1.

Полученная таким образом запись является шестнадцатеричной записью искомого числа  $R$ .

Например, возьмём число 91. Строим шестнадцатеричную запись числа: 5B. Цифра  $B$  в нём одна – нечётное количество, значит, единицу дописываем справа. Итоговое число 5B1 переводим в десятичную систему – 1457. Число 1457 и является результатом работы алгоритма.

Определите количество натуральных чисел  $N$ , для которых результатом выполнения алгоритма может стать двухзначное число.

Ссылка на видео-разбор с таймингом: [https://vk.com/video-205546952\\_456241250?t=0h30m15s](https://vk.com/video-205546952_456241250?t=0h30m15s)



## Решение

Для 16-теричной системы не надо писать отдельную функцию, потому, что мы можем использовать либо функцию `hex`, или `f`-строку.

Цикл перебора выполняется для значений `n` от 1 до 99. В ходе каждого шага цикла получаем шестнадцатеричное представление числа `n` через `f`-строку.

Затем анализируем количество цифр "b" в этом представлении. Если количество этих цифр четное, добавляем единицу слева, если нечетное — справа.

После этого преобразуем полученное число обратно в десятичное представление и проверяем, является ли оно двузначным. Если да, сохраняем текущее значение `n`.

```
m = []
for n in range(1,100):
    b = f'{n:x}'
    if b.count('b')%2==0:
        b = '1' + b
    else:
        b = b + '1'
    r = int(b,16)
    if 10<=r<100:
        m.append(n)
print(len(m))
```

Результат работы программы:

14

По завершении цикла программа выводит количество найденных подходящих значений `n`, нашлось 14 таких чисел.

Ответ: 14

Telegram: @fast\_ege

## DZ\_5\_14

### Задача № 14(1516)

Автомат обрабатывает натуральное число `N` по следующему алгоритму:

- 1) Строится двоичная запись числа `N`.
- 2) Из записи удаляются все нули.

3) Полученное число переводится в десятичную запись и выводится на экран.

Сколько разных значений будет показано на экране автомата при последовательном вводе всех натуральных чисел от 100 до 1000?

Ссылка на видео-разбор с таймингом: [https://vk.com/video-205546952\\_456241250?t=0h32m45s](https://vk.com/video-205546952_456241250?t=0h32m45s)

### Решение

Для решения задачи можно использовать следующую последовательность действий:

1. Объявляется множество  $m$ , которое позволит хранить уникальные результаты без дубликатов.
2. Организуем цикл по всем числам в указанном диапазоне от 100 до 1000.

Для каждого числа  $n$ :

- Строим его двоичное представление.
- Удаляем из этой записи все нули путем замены их на пустую строку.
- Преобразуем оставшуюся строку единиц обратно в десятичный формат.
- Добавляем результат в множество  $m$ .

После завершения цикла, размер множества  $m$  покажет количество уникальных значений, которые могут быть получены в результате выполнения данного алгоритма.

```
m = []
for n in range(100,1001):
    b = f'{n:b}'
    b = b.replace('0','')
    r = int(b,2)
    m.append(r)
print(len(set(m)))
```

Результат работы программы:

9

Результат работы программы показывает, что существует 9 различных значений, которые могут быть выведены на экран при выполнении описанного в условии задачи алгоритма.

Ответ: 9

Telegram: @fast\_ege

**Задача № 15(1514)**

Автомат обрабатывает натуральное число  $N$  по следующему алгоритму::

- 1) Строится двоичная запись числа  $N$ .
  - 2) Запись «переворачивается», то есть читается справа налево. Если при этом появляются ведущие нули, они отбрасываются.
  - 3) Полученное число переводится в десятичную запись и выводится на экран.
- Какое наибольшее число, не превышающее 100, после обработки автоматом даёт результат 9?

Ссылка на видео-разбор с таймингом: [https://vk.com/video-205546952\\_456241250?t=0h35m0s](https://vk.com/video-205546952_456241250?t=0h35m0s)

*Решение*

Мы рассматриваем диапазон чисел от 1 до 100 включительно.

Для начала мы строим двоичное представление числа и затем переворачиваем его. Переворот строки можно осуществить с помощью среза. В Python это делается следующим образом: `b = b[::-1]`. Этот срез читает строку символов в обратном порядке, таким образом, мы меняем порядок цифр в числе.

Что касается ведущих нулей, то их не требуется специально удалять, поскольку функция `int` сама игнорирует их при преобразовании строки в целое число. Когда мы снова переводим перевернутое число в десятичное представление, любые начальные нули будут проигнорированы.

Затем проверяем, равен ли результат числу 9. Если да, то выводим исходное число `n`, при котором это произошло. Запуская программу, мы получаем результат.

```
for n in range(1,101):  
    b = f'{n:b}'  
    b = b[::-1]  
    r = int(b,2)
```

```
if r==9:  
    print(n)
```

Результат работы программы:

9  
18  
36  
72

В результате выполнения программы выводятся следующие числа: 9, 18, 36 и 72. Из них максимальное значение — 72.

Ответ: 72

Telegram: @fast\_ege

## DZ\_5\_16

### Задача № 16(564)

Автомат обрабатывает натуральное число  $N < 128$  по следующему алгоритму:

1. Строится восьмибитная двоичная запись числа  $N$ .
2. Инвертируются разряды исходного числа (0 заменяется на 1, 1 на 0).
3. К полученному двоичному числу прибавляют единицу.
4. Полученное число переводится в десятичную систему счисления.

Для какого числа  $N$  результат работы алгоритма равен 153?

Какое наибольшее число, не превышающее 100, после обработки автоматом даёт результат 9?

Ссылка на видео-разбор с таймингом: [https://vk.com/video-205546952\\_456241250?t=0h36m55s](https://vk.com/video-205546952_456241250?t=0h36m55s)

### Решение

Рассмотрим следующую задачу. Алгоритм обработки натурального числа  $n$ , которое меньше 128, состоит из следующих шагов:

1. Формируется 8-битная двоичная запись числа  $n$ . Важно помнить, что она должна содержать ровно восемь цифр.
2. Разряды исходного числа инвертируются: все нули заменяются на единицы, а единицы — на нули.

3. К полученному двоичному числу добавляется единица.
4. Результат преобразуется обратно в десятичную систему счисления.

Необходимо определить, при каком значении  $n$  результат работы алгоритма будет равен 153.

Для решения задачи необходимо перебрать значения  $n$  от 1 до 127 включительно.

1. Построение 8-битной двоичной записи числа  $n$  : используется форматирование строки, обеспечивающее наличие ровно восьми символов, при необходимости добавляются ведущие нули.
2. Инвертирование разрядов: поскольку простая замена всех нулей на единицы приведет к потере информации о положении изначальных нулей, применяется промежуточная операция замены нулей на двойки, затем единиц на нули, и наконец, оставшихся двоек на единицы.
3. Добавление единицы к полученному двоичному числу удобнее выполнять после перевода его в десятичное представление, так как арифметические операции над строками невозможны.
4. Если результат равен 153, выводится соответствующее значение  $n$ .

```
for n in range(1,128):  
    b = f'{n:08b}'  
    b = b.replace('0','2').replace('1','0').replace('2','1')  
    r = int(b,2) + 1  
    if r==153:  
        print(n)
```

Результат работы программы:

103

Ответ: 103

Telegram: @fast\_ege