

DZ_52_1

Задача № 1 (688)

Автомат обрабатывает натуральное число N ($0 \leq N \leq 255$) по следующему алгоритму:

- 1) Строится восьми битная двоичная запись числа N .
 - 2) Удаляются средние 4 цифры.
 - 3) Полученное число переводится в десятичную запись и выводится на экран.
- Какое наибольшее число, меньшее 110, после обработки автоматом даёт результат 7?

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241255?t=0h0m0s

Решение

Для решения этой задачи переберём все возможные значения от 0 до 109 включительно:

Для каждого числа n из диапазона $[0, 109]$:

1. Преобразуем n в 8-битное двоичное представление.
2. Удалим 4 центральных бита.
3. Объединим оставшиеся биты и перевести их обратно в десятичный формат.
4. Проверим, равно ли полученное значение числу 7. Если да, выведем на печать текущее исходное число n .

После завершения перебора выберем максимальное число среди тех, которые при обработке дали результат 7.

```
for n in range(110):
    b = f'{n:08b}'
    b = b[:2] + b[-2:]
    r = int(b, 2)
    if r==7:
        print(n)
```

Результат работы программы:

67
71
75
79
83
87
91
95
99
103
107

Ответ: 107

Telegram: @fast_ege

Задача №2 (612)

Автомат обрабатывает натуральное число N по следующему алгоритму.

- 1 Строится двоичная запись числа N .
- 2 Удаляется первая слева единица и все следующие непосредственно за ней нули. Если после этого в числе не остаётся цифр, результат этого действия считается равным нулю.
- 3 Полученное число переводится в десятичную запись.
- 4 Новое число вычитается из исходного, полученная разность выводится на экран.

Пример. Дано число $N = 11$ Алгоритм работает следующим образом.

- 1 Двоичная запись числа N : 1011
- 2 Удаляется первая единица и следующий за ней ноль: 11
- 3 Десятичное значение полученного числа 3
- 4 На экран выводится число $11 - 3 = 8$

Сколько разных значений будет показано на экране автомата при последовательном вводе всех натуральных чисел от 10 до 1000?

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241255?t=0h2m30s

Решение

В этой задаче требуется подсчитать различные значения, которые могут появиться на экране после обработки всех натуральных чисел от 10 до 1000 включительно.

Для каждого числа нужно сделать следующее:

1. Представить его в двоичном виде.
2. Убрать первую единицу и следующие за ней нули (если они есть).
3. Перевести остаток обратно в десятичное число.
4. Вычеркнуть это значение из исходного числа.

При преобразовании стоит учесть, что Python автоматически убирает ведущие нули при переводе строки в целое число. Поэтому достаточно убрать только первую единицу из двоичного представления числа.

Изучив результаты, можно увидеть, что среди них присутствуют степени двойки. Некоторые из них появляются несколько раз. Т.к. для решения задачи требуется найти количество уникальных значений, мы будем использовать `set()`, то есть множество.

Используя множество, мы избавимся от дублей и получим финальный набор разных значений.

```
d = set()
for n in range(10, 1001):
    b = f'{n:b}'
    b = b[1:]
    r = int(b, 2)
    r = n - r
    d.add(r)
print(d)
```

Результат работы программы:

{32, 64, 128, 256, 512, 8, 16}

В результате работы программы мы получили 7 различных значений.

Ответ: 7

Telegram: @fast_ege

DZ_52_3

Задача № 3 (1332)

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

1. Число N переводим в двоичную запись
2. Инvertируем все биты числа кроме первого
3. Переводим в десятичную запись
4. Складываем с исходным числом N

Полученное число является искомым числом R . Укажите наименьшее нечетное число N , для которого результат работы данного алгоритма больше 99. В ответе это число запишите в десятичной системе счисления.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241255?t=0h5m55s

Решение

Для поиска решения перебираем различные значения n от 1 до 100. Для каждого числа получаем его двоичное представление $f'\{n:b\}'$.

Затем необходимо инvertировать все биты, кроме первого. Создаем переменную $b1$, которая содержит все цифры двоичного представления числа n , исключая первую цифру. После этого производим инверсию всех цифр в $b1$: заменяем 0 на 1 и наоборот.

Восстанавливаем полное двоичное представление числа, объединяя неизмененный первый бит и инvertированную часть $b1$. Преобразуем полученное значение обратно в десятичный формат, чтобы получить число r .

Суммируем r с исходным значением n . Если результат превышает 99, выводим найденное значение n .

```
for n in range(1,100):
    b = f'{n:b}'
    b1 = b[1:]
    b1 = b1.replace('0','2').replace('1','0').replace('2','1')
    b = b[0]+b1
    r = int(b,2)
    r = r+n
    if n%2!=0 and r>99:
        print(n)
```

Результат работы программы:

65

67

69

...

Минимальное значение равно n

Ответ: 65

Telegram: @fast_ege

DZ_52_4

Задача № 4 (1505)

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

1. Строится двоичная запись числа N .
2. Каждый разряд этой записи заменяется двумя разрядами по следующему правилу: если в разряде стоит 0, то вместо него пишется 01; если в разряде стоит 1, то 1 заменяется на 10.

Например, двоичная запись 1001 числа 9 будет преобразована в 10010110.

Полученная таким образом запись (в ней в 2 раза больше разрядов, чем в записи исходного числа N) является двоичной записью искомого числа R .

Укажите максимальное число R , меньшее 256, которое может являться результатом работы данного алгоритма. В ответе это число запишите в десятичной системе.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241255?t=0h8m55s

Решение

Для решения будем перебирать числа n от 1 до 200. И, вместо того чтобы напрямую изменять нули и единицы, мы временно заменим нули на двойки, а единицы на тройки. Таким образом, когда мы начнем производить замену, 2 (которые изначально были нулями) будут превращаться в 01, но не затронут

оставшиеся 3 (единицы). Исходные единицы останутся помеченными как 3, поэтому они не будут перепутаны с новыми символами. Затем мы просто заменим 3 на 10.

Таким образом, изменяться будут только первоначальные цифры нашей двоичной строки. Новые символы, появившиеся в результате преобразования, остаются нетронутыми, так как мы работаем исключительно с 2 и 3, которые соответствуют исходным цифрам.

```
m = []
for n in range(1,200):
    b = f'{n:b}'
    b = b.replace('0','2').replace('1','3')
    b = b.replace('2','01').replace('3','10')
    r = int(b,2)
    if r<256:
        m.append(r)
print(max(m))
```

Результат работы программы:

170

Самое большое число r , которое подходит под все условия, равно 170.

Ответ:170

Telegram: @fast__ege

DZ_52_5

Задача №5 (7693)

Автомат обрабатывает натуральное число N по следующему алгоритму.

1. В шестеричной записи числа N дублируется последняя цифра
2. Полученное число переводится в двоичную систему счисления.
3. Искомое R - сумма цифр в конечной версии числа.

Пример.

$N = 35$

1. $35_{10} = 55_6$. $'55' + '5' = '555'$

2. $555_6 = 11010111_2$

3. $R = 1 + 1 + 0 + 1 + 0 + 1 + 1 + 1 = 6$

Напишите максимальное число N , не превышающее 105, для которого $R = 18$.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241255?t=0h12m0s

Решение

Для решения задачи будем перебирать значения n в диапазоне от 1 до 100 000, то есть до 105. Для начала переведем число в шестеричную систему.

Поскольку Python не предоставляет встроенной функции для этого, создадим свою собственную функцию для перевода числа в шестеричную систему.

Функция работает так: пока $x > 0$, она добавляет остаток от деления x на 6 к строке результата, а затем делит x нацело на 6. После завершения цикла функция возвращает строку.

Теперь, когда у нас есть шестеричная запись числа, обозначенная как $b6$, мы можем удвоить последнюю цифру. Это делается путем добавления последней цифры к самому числу: $b6 + b6[-1]$.

Далее необходимо перевести полученное число в двоичную систему. Для этого сначала переведем его в десятичную систему, а затем используем форматированную строку для преобразования в двоичную. Подсчитываем единицы в полученной двоичной записи, чтобы получить r .

Если $r = 18$, выводим соответствующее значение n .

```
def cc6(x):
    s = ''
    while x>0:
        s = str(x%6+s)
        x = x//6
    return s
for n in range(1,100000):
    b6 = cc6(n)
    b6 = b6 + b6[-1]
    r = int(b6,6)
    r = f'{r:b}'
    r = r.count('1')
    if r==18:
        print(n)
```

Результат работы программы:

86015

87359

В ответ запишем максимальное из полученных чисел.

Ответ: 87359

Telegram: @fast_ege

DZ_52_6

Задание №6 (2851)

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

1. Строится двоичная запись числа N .
2. Вычисляется количество единиц, стоящих на чётных местах в двоичной записи числа N без ведущих нулей, и количество нулей, стоящих на нечётных местах. Места отсчитываются слева направо (от старших разрядов к младшим, начиная с единицы).
3. Результатом работы алгоритма становится модуль разности полученных двух чисел.

Пример. Дано число $N = 39$. Алгоритм работает следующим образом:

1. Строится двоичная запись: $39_{10} = 100111_2$.
2. Выделяем единицы на чётных и нули на нечётных местах: 100111.

На чётных местах стоят две единицы, на нечётных – один ноль.

3. Модуль разности равен 1. Результат работы алгоритма $R = 1$.

При каком наименьшем N в результате работы алгоритма получится $R = 5$?

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241255?t=0h14m50s

Решение

Для числа n строится его двоичная запись. Затем определяется количество единиц, находящихся на четных позициях этой записи, а также количество нулей, расположенных на нечетных позициях. Позиции считаются слева направо, начиная с первой.

Рассмотрим метод получения четных и нечетных позиций. Нумерация позиций выглядит следующим образом: 1, 2, 3, 4, 5, 6, 7, 8. Чтобы извлечь нечетные позиции, используется операция взятия элементов через шаг 2. Таким образом, получаем последовательность 1, 3, 5, 7.

Аналогично для четных позиций: выбираем элементы, начиная со второй позиции, что дает последовательность 2, 4, 6, 8.

Запишем цикл для чисел от 2 до 10000. Для каждого числа строим его двоичное представление. Затем определяем количество единиц на четных позициях и количество нулей на нечетных позициях. Вычисляем абсолютную разницу между этими значениями. Если разница равна 5, выводим соответствующее число.

```
for n in range(2,10000):
    b = f'{n:b}'
    k1 = b[1::2].count('1')
    k2 = b[::2].count('0')
    r = abs(k2-k1)
    if r==5:
        print(n)
```

Результат работы программы:

1023

1024

2047

2048

...

Ответ: 1023

Telegram: @fast_ege

DZ_52_7

Задание № 7(971)

Автомат получает на вход четырехзначное число. По этому числу строится новое число по следующим правилам.

1. Умножаются первая и вторая, а также третья и четвертая цифры исходного числа.
 2. Полученные два числа записываются друг за другом в порядке возрастания (без разделителей).
- Пример. Исходное число: 5431. Произведения: $5 * 4 = 20$; $3 * 1 = 3$. Результат: 320.

Укажите максимальное число, в результате обработки которого, автомат выдаст число 1214.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241255?t=0h18m0s

Решение

Необходимо найти максимальное число, для которого программа выдаст результат 1214. Для поиска такого числа мы будем перебирать все четырёхзначные числа в диапазоне от 1000 до 9999.

Чтобы работать с отдельными цифрами числа, мы преобразуем его в строку, а затем разбиваем эту строку на отдельные символы, каждый из которых превращаем обратно в целое число. В результате у нас получается список из четырёх цифр исходного числа.

Затем мы вычисляем произведение первой и второй цифр, а также третьей и четвёртой. Далее полученные результаты размещаются друг за другом в порядке возрастания. Если первое полученное число меньше второго, то они просто объединяются в одну строку; в противном случае второе число становится первым, а первое — вторым.

Если результирующая строка равна "1214", она выводится на экран.

```
for n in range(1000,10000):
    d = [int(x) for x in str(n)]
    a1 = d[0]*d[1]
    a2 = d[2]*d[3]
    if a1<a2:
        r = str(a1+str(a2))
    else:
```



```
r = str(a2+str(a1))  
if r=='1214':  
    print(n)
```

Результат работы программы:

2627

2672

2726

2734

2743

2762

3427

3472

4327

4372

6227

6272

7226

7234

7243

7262

После запуска программы видно, что максимальное число, которое соответствует условиям задачи, это 7262. Это происходит потому, что произведение первых двух цифр равно 12, а последних двух — 14, и они расположены в правильном порядке.

Ответ: 7262

Telegram: @fast_ege

Задание №8 (972)

Автомат получает на вход трехзначное число. По этому числу строится новое число по следующим правилам.

1. Умножаются первая и вторая, а также вторая и третья цифры исходного числа.
2. Полученные два числа записываются друг за другом в порядке убывания (без разделителей).

Пример. Исходное число: 543. Произведения: $5 * 4 = 20$; $4 * 3 = 12$. Результат: 2012.

Укажите максимальное число, в результате обработки которого, автомат выдаст число 240.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241255?t=0h20m40s

Решение

Нам нужно определить максимальное число, для которого результат работы программы будет 240, следовательно, один из результатов действий должен быть равен нулю. Проверим все трёхзначные числа от 100 до 999. Для этого создадим список цифр каждого числа. Эти цифры будут извлекаться из строки, представляющей число n , после чего они преобразуются в числовой формат.

Теперь выполним операции над этими цифрами. $a1$ будет результатом умножения $d0$ (первая цифра) на $d1$ (вторая цифра). Аналогично, $a2$ получится путём умножения $d1$ на $d2$. После этого два полученных числа соединятся в одно по принципу убывания: если $a1$ больше $a2$, то оно запишется первым, иначе – вторым.

```
for n in range(100,1000):
    d = [int(x) for x in str(n)]
    a1 = d[0]*d[1]
    a2 = d[1]*d[2]
    if a1>a2:
        r = str(a1+str(a2))
    else:
        r = str(a2+str(a1))
    if r=='240':
        print(n)
```

Результат работы программы:

380

460

640

830

При запуске программы видно, что максимальное число, дающее результат 240, это 830. Это логично, ведь $8*3=24$, а последняя цифра равна нулю.

Ответ: 830

Telegram: @fast_ege

DZ_52_9

Задание №9(206)

Автомат обрабатывает трёхзначное натуральное число N по следующему алгоритму.

1. Из цифр, образующих десятичную запись N , строятся наибольшее и наименьшее возможные двузначные числа (числа не могут начинаться с нуля).

2. На экран выводится разность полученных двузначных чисел.

Пример. Дано число $N = 351$. Алгоритм работает следующим образом.

1. Наибольшее двузначное число из заданных цифр – 53, наименьшее – 13.

2. На экран выводится разность $53 - 13 = 40$.

Чему равно количество чисел N на отрезке $[800; 900]$, в результате обработки которых на экране автомата появится число 30?

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241255?t=0h23m5s

Решение

Рассмотрим числа в указанном диапазоне. Из каждого числа необходимо получить строку, чтобы иметь возможность оперировать его цифрами. Далее следует рассмотреть все возможные пары цифр, образующие двузначные числа. Всего таких пар существует 6:

$(S_0 + S_1),$

$(S_0 + S_2),$

$(S_1 + S_0),$

$(S_1 + S_2),$

$(S_2 + S_0),$

$(S_2 + S_1).$

Из примера рассмотренного в условии задачи видно, что одинаковые цифры не могут быть использованы более одного раза при формировании двузначных чисел. Следовательно, необходимо преобразовать эти строки в целые числа и исключить те из них, которые начинаются с нуля (то есть однозначные). Для этого остаются только те числа, которые больше или равны

10. Затем определяется разница между максимальным и минимальным значениями среди полученных двузначных чисел. Эта разность

должна быть равна 30. Анализируя различные комбинации, можно найти следующие числа, удовлетворяющие этому условию.

```
for n in range(800, 901):
    s = str(n)
    a = [s[0]+s[1], s[0]+s[2], s[1]+s[0], s[1]+s[2], s[2]+s[0], s[2]+s[1]]
    a = [int(x) for x in a]
    a = [x for x in a if x >= 10]
    r = max(a) - min(a)
    if r == 30:
        print(n)
```

Результат работы программы:

855

856

857

858

865

869

875

885

896

Мы получили 9 чисел в результате обработки которых на экране автомата появится число 30

Ответ: 9

Telegram: @fast_ege

DZ_52_10

Задание №10 (3754)

Алгоритм получает на вход натуральное число $N \geq 10$ и строит по нему новое число R следующим образом:

1 Все пары соседних цифр в десятичной записи N рассматриваются как двузначные числа (возможно, с ведущим нулём.

2 Из списка полученных на предыдущем шаге двузначных чисел выделяются наименьшее и наибольшее.

3 Результатом работы алгоритма становится сумма найденных на предыдущем шаге двух чисел.

Пример. Дано число $N = 2022$ Алгоритм работает следующим образом:

1 В десятичной записи выделяем двузначные числа: 20, 02, 22

2 Наименьшее из найденных чисел 02, наибольшее 22

3 $02 + 22 = 24$

Результат работы алгоритма $R = 24$

При каком наименьшем N в результате работы алгоритма получится $R = 137$?

Ссылка на видео-разбор с таймингом: <https://vk.com/video-205546952456241255?t=0h27m5s>

Решение

Есть смысл начинать начать сразу с трёхзначных чисел, так как для двузначных чисел существует всего одна пара, а значит, наибольшее и наименьшее числа совпадают. Таким образом, получить 137 невозможно, поскольку сумма двух одинаковых чисел всегда будет чётной.

Поэтому рассматриваем трёхзначные числа от 100 до 999. Возьмём строку, представляющую это число, и рассмотрим его соседние цифры. Можно сразу преобразовать их в целые числа: $s_0 + s_1$ и $s_1 + s_2$.

Теперь у нас есть две пары соседних цифр: $s_0 + s_1$ и $s_1 + s_2$. Нужно найти сумму максимального и минимального значений среди них. Если эта сумма равна 137, выводим соответствующее число n .

```
for n in range(100,1000):
    s = str(n)
    a = [int(s[0]+s[1], int(s[1]+s[2]))]
    r = max(a)+min(a)
    if r==137:
        print(n)
```

Результат работы программы:

398
489
670
761
852
943

Выведенные результаты показывают, что подходящим числом является 398, где $39 + 98 = 137$. Это минимальный возможный вариант.

Ответ : 398

