**Задание № 25**

**Р-03.** *Найдите все натуральные числа, принадлежащие отрезку [77 777 777; 88 888 888], у которых ровно пять различных нечётных делителей (количество чётных делителей может быть любым). В ответе перечислите найденные числа, справа от каждого числа запишите его наименьший нечётный делитель, не равный 1.*

Первый способ – решение «в лоб».

1) Перебираем числа из диапазона, для каждого числа считаем количество нечетных делителей (не забываем прибавить 1 – единицу).

2) не забудем, что вывести нужно не только число с 5-ю нечетными делителями, но и минимальный делитель, запомним его.

3) если у числа есть 3 делителя нечетных, выводим.

**Программы для справки и использования:**

1) количество чисел, удовлетворяющих условию:

**count = 0**

**for n in range(a, b+1):**

 **if *условие выполнено*:**

 **count += 1**

 2) делится – не делится:

**if n % d == 0:**

 **print("Делится")**

**else: print("Не делится")**

3) сконструируйте часть программы, которая считает количество нечетных делителей и выводит число, а также наименьший делитель для чисел, у которых количество делителей равно 5.

Можно все текущие делители числа записывать в массив и выводить только элемент с индексом 0:

**divs = []**

**for d in range(1,n+1): # перебор всех возможных делителей**

 **if n % d == 0: # здесь еще условие будет**

 **divs.append(d) # то добавили его в массив**

Сколько времени программа работает?

Ответы: 77900162 79 78074896 47

 78675968 7 80604484 67

 81920000 5 84934656 3

 85525504 17 88529281 97

Второй способ – предварительный анализ нужен, чтобы сократить время работы программы.

План решения:

1) Заметим, что если число a имеет 1 нечетное простое число b своим делителем, то нечетных делителей у него всего 2: единица и b. Например, число 10=1\*2\*5

2) если нечетных простых делителей 2, например, как у числа 30 (делители 1, 2, **3, 5**, 6, 10, 15, 30), то делителей становится 3+1 =4 т.е. еще один нечетный делитель появляется их произведения 2-х простых нечетных. Далее, при увеличении разных простых делителей количество нечетных делителей будет равно 7, что больше, чем нам нужно.

Заметим, что четные делители не влияют на кол-во нечетных, поэтому не будем их рассматривать.

3) Рассмотрим вариант, когда нечетный делитель один, но он повторяется несколько раз: 9=1 и 2 раза 3: количество нечетных делителей: 3(1, 3, 9) .

Или 3 раза: 27= 1 и 3 раза 3: количество делителей: 4 (1, 3, 9, 27)

4 раза: 81: 1 и 4 раза 3: 5 (1, 3, 9, 27, 81)

Итак, **правило: для того, чтобы у числа было ровно 5 разных простых делителей нужно, чтобы в его разложение на простые множители входило только 1 нечётное простое число с кратностью 4**. Само число имеет такую структуру: N=1\*2k \*b4

Пример: 5\*5\*5\*5\*4=2500. Его нечетные делители: 1, 5,25, 125, 625,

4) Задача свелась к тому, чтобы перебрать числа из отрезка и, убрав из их разложения на простые множители 2k, определить является ли то, что осталось четвёртой степенью простого числа.

Для этого нужен будет цикл while для деления на 2, пока это возможно.

Число, которое осталось, проверим: это четвертая степень целого числа? Можно воспользоваться from math import sqrt и округлением round

Осталось проверить полученное число на простоту.

5) Наименьшим простым нечётным делителем, отличным от единицы, будет это простое число. Для определения простоты числа воспользуемся вариантом функции isPrime() без вещественных чисел

# функция для определения простоты числа

def isPrime( x ):

 if x <= 1: return False

 d = 2

 while d\*d <= x:

 if x % d == 0:

 return False

 d += 1

 return True

6) Дальше вы должны написать программу, запустить ее, какая работает быстрее?