

## A more or less formal solution

The problem asks to find out  $x$  digit numbers for  $x = (3, \dots, 7)$  whose last 3 digits are 388 (and count them).

So for every number  $x$  it's easy to see that in the result of:

$$n \times 10^{(\text{number of digits in } n)} + n$$

ends with the number  $n$  and this result is a multiple by  $n$ .

*i.e.:*

$$8 \times (10 + 1) = 8 \times 10 + 8 \times 1 = 80 + 8 = 88$$

$$76 \times 101 = 7600 + 76 = 7676$$

In particular:

$$388 \times 1001 = 388388$$

So we can formulate it like this:

$$388 \times k < 10000000$$

From where we need all the  $k$ 's that verify that  $388 \times k$  ends in 388.

If we divide by 388 and round up we can obtain the range of values that  $k$  will take. The result of the division is 25773, so  $388 \times k < 10000000$  if  $k < 25773$ .

Now, as the prime decomposition of 388 is  $2 \times 2 \times 97$ , we can write 388 as  $4 \times 97$ .

So decomposing 388 and 1000 we can reformulate  $388 \times 1000 = (4 \times 97) \times (4 \times 250)$  so every 250  $k$ 's we have a multiple of 388 that ends with three zeros, and if we add 388 to that number, we have found one of the numbers we are looking for!.

So now we see how many of this numbers are in the valid range:

$$\frac{25773}{250} = 103$$

So  $388 \times (250 \times i)$  for  $i = (1, \dots, 103)$  are all valid numbers.

To end up we need to count the solution obtained by:

$$388 \times (250 \times 0) = 388$$

that we weren't counting (Note that  $388 \times (250 \times 1) = 97000$ ).

Adding up, in total there's **104** numbers that are seven digits or less whose last three digits are 388 in that order.

## Python code to check the solution

To check out the number obtained before I've written this piece of code as concise as possible:

```
1 print (len([ x for x in \
2         map(lambda n: n[-3:], \
3             map(str, \
4                 [y * 388 for y in range(1, int(10000000/388))])) \
5             if '388' in x ]))
```

Basically what I'm doing is:

- As the smallest 8 number digit is 10000000 we can find out how many multiples of 388 that are less than 10000000 exist by dividing both numbers (line 4).
- Creating a list of all multiples of 388 in the range 388 and 10000000 (line 4).
- Using the map function to convert all the elements of this list (which are integers) to strings (line 3).
- Defining a lambda function which given an element, returns it's last 3 characters (line 2).
- Using the map function with the previous mentioned lambda function on the list resulting of applying the first map function mentioned before (line 2).
- Extracting elements from the resulting list of the second map function application, which satisfy that the string '388' is a part of this element (line 1).
- We can easily note that this returns a list of elements which are all completely equal strings '388'. So we count how many elements are on this list and print the result (line 1).

This is actually just one line of code.