Multiple Surprises NRICH Solution

I was able to leverage my python programming skills to help solve this problem. I used a simple while loop with a counter x, so that I could get x number of lists which satisfy

- the first is a multiple of 1
- the second is a multiple of 2
- the third is a multiple of 3
- the fourth is a multiple of 4
- the fifth is a multiple of 5
- the sixth is a multiple of 6
- the seventh is a multiple of 7
- the eighth is a multiple of 8
- the ninth is a multiple of 9
- the tenth is a multiple of 10?

I noticed that the best way to make the smallest number of iterations was by looking at the multiple of 10, and going from there as of the numbers required to satisfy, it has the least number of multiples in a given range. Hence, I decided that each iteration will take a multiple of 10 and then check to see if it satisfies the conditions. If it does, it is outputted and given. By running this code alone, I get the following four lists:

- 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
- 2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530
- 5041, 5042, 5043, 5044, 5045, 5046, 5047, 5048, 5049, 5050
- 7561, 7562, 7563, 7564, 7565, 7566, 7567, 7568, 7569, 7570

By simply increasing the value of x, I am able to get more lists:

- 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
- 2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530
- 5041, 5042, 5043, 5044, 5045, 5046, 5047, 5048, 5049, 5050
- 7561, 7562, 7563, 7564, 7565, 7566, 7567, 7568, 7569, 7570
- 10081, 10082, 10083, 10084, 10085, 10086, 10087, 10088, 10089, 10090
- 12601, 12602, 12603, 12604, 12605, 12606, 12607, 12608, 12609, 12610
- 15121, 15122, 15123, 15124, 15125, 15126, 15127, 15128, 15129, 15130
- 17641, 17642, 17643, 17644, 17645, 17646, 17647, 17648, 17649, 17650
- 20161, 20162, 20163, 20164, 20165, 20166, 20167, 20168, 20169, 20170
- 22681, 22682, 22683, 22684, 22685, 22686, 22687, 22688, 22689, 22690
- 25201, 25202, 25203, 25204, 25205, 25206, 25207, 25208, 25209, 25210
- 27721, 27722, 27723, 27724, 27725, 27726, 27727, 27728, 27729, 27730
- 30241, 30242, 30243, 30244, 30245, 30246, 30247, 30248, 30249, 30250
- 32761, 32762, 32763, 32764, 32765, 32766, 32767, 32768, 32769, 32770
- 35281, 35282, 35283, 35284, 35285, 35286, 35287, 35288, 35289, 35290
- 37801, 37802, 37803, 37804, 37805, 37806, 37807, 37808, 37809, 37810

- 40321, 40322, 40323, 40324, 40325, 40326, 40327, 40328, 40329, 40330
- 42841, 42842, 42843, 42844, 42845, 42846, 42847, 42848, 42849, 42850
- 45361, 45362, 45363, 45364, 45365, 45366, 45367, 45368, 45369, 45370
- 47881, 47882, 47883, 47884, 47885, 47886, 47887, 47888, 47889, 47890

Doing this task by hand proves difficult hence why in my opinion, making a program is the most efficient and effective solution.

Python

```
x = 4
count = 10

while x != 0:
  if (count - 1) %9 == 0 and (count-2)%8==0 and (count-3)%7==0 and
(count-4)%6==0 and (count-5)%5==0 and (count-6)%4==0 and (count-7)%3==0 and
(count-8)%2==0:
    f = ''
    for i in range(count-9, count+1):
      f += str(i)
      if i != count:
        f += ", "
    print(f)
    x -= 1
  count += 10
```