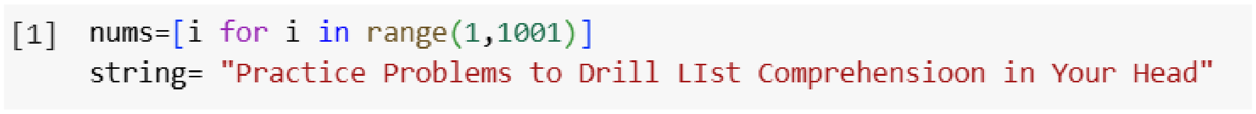
Lab02-Report (Exercise):

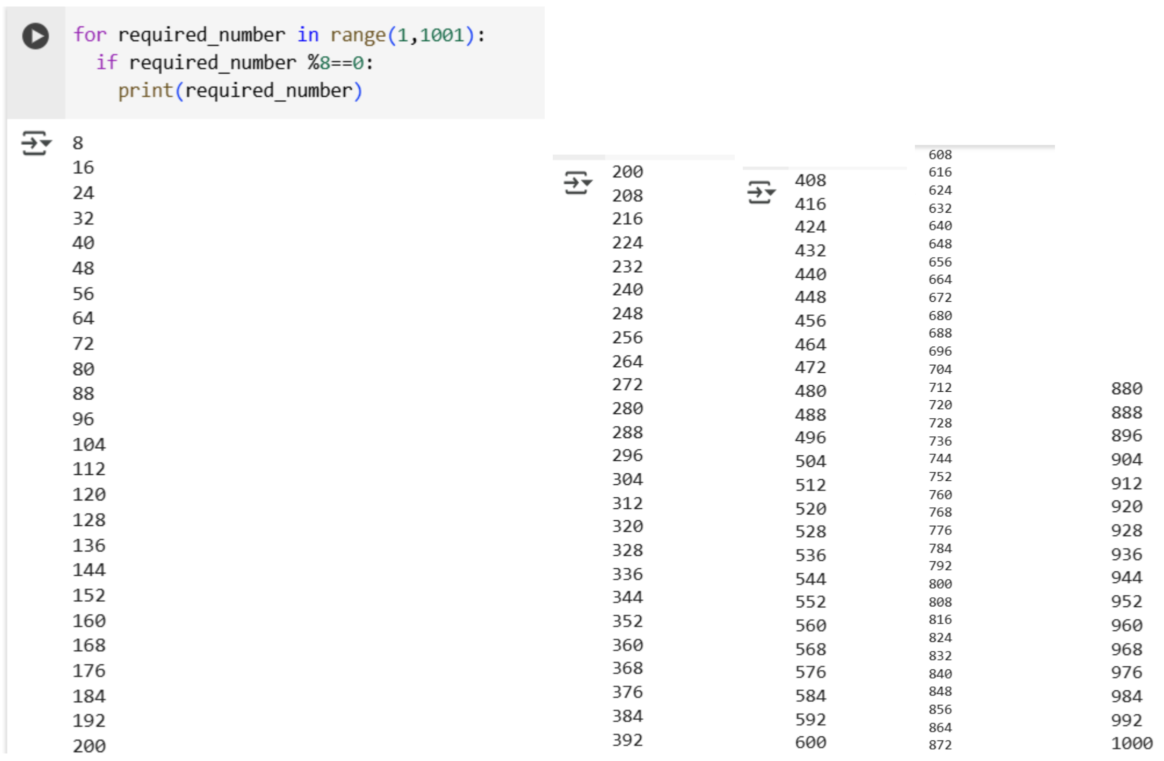
This part has been implemented at the beginning of the code in colab to implement it in each code.



```
[1] nums=[i for i in range(1,1001)]
    string= "Practice Problems to Drill LIst Comprehensioon in Your Head"
```

Question-01:

To find all the numbers from 1-1000 divisible by 8 a “For Loop” with “If” condition where the %8 condition has been implemented.

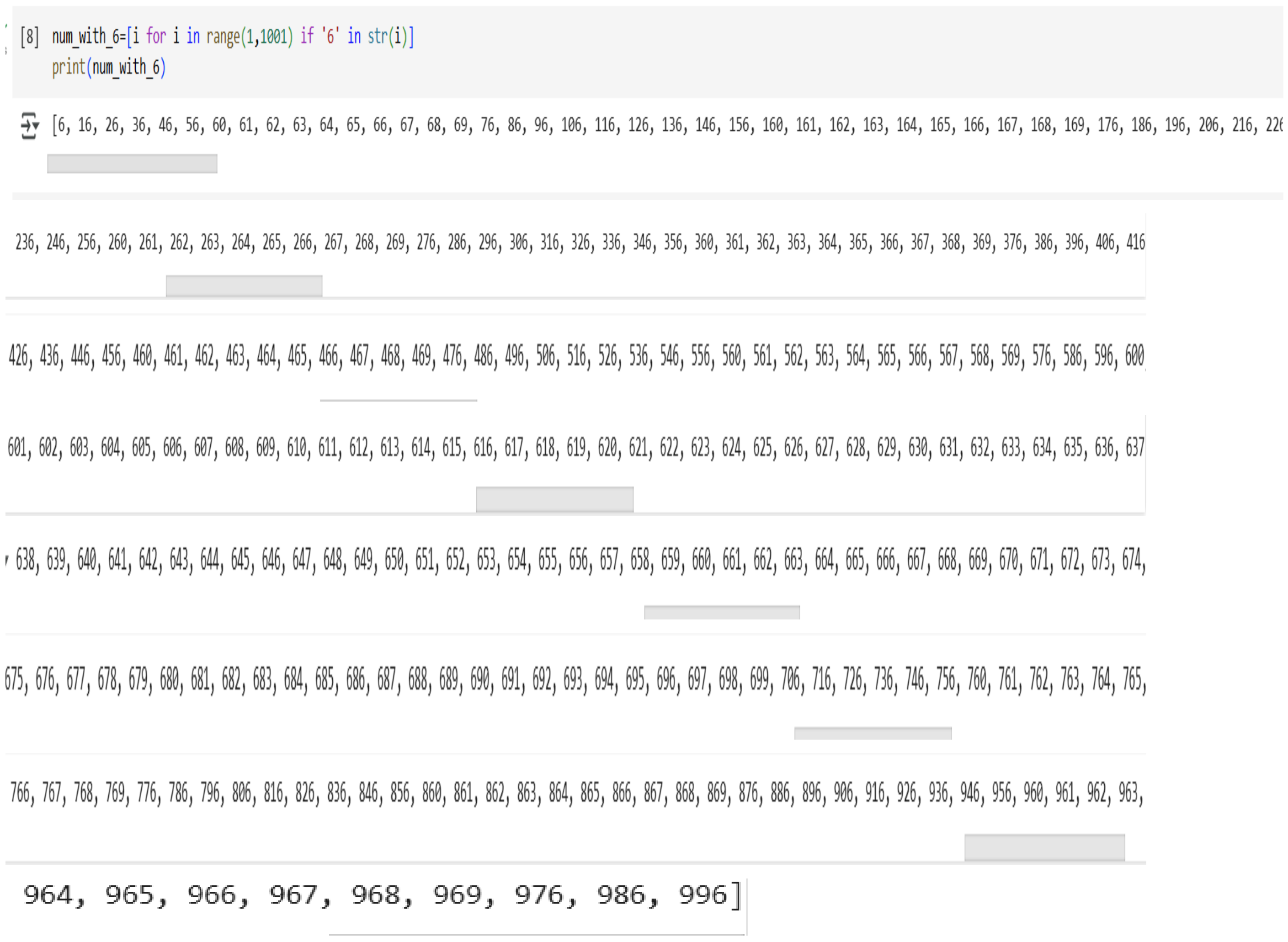


```
for required_number in range(1,1001):
  if required_number %8==0:
    print(required_number)
```

```
8        200      408      608      880
16       208      416      616      888
24       216      424      624      896
32       224      432      632      904
40       232      440      640      912
48       240      448      648      920
56       248      456      656      928
64       256      464      664      936
72       264      472      672      944
80       272      480      680      952
88       280      488      688      960
96       288      496      696      968
104      296      504      704      976
112      304      512      712      984
120      312      520      720      992
128      320      528      728      1000
136      328      536      736
144      336      544      744
152      344      552      752
160      352      560      760
168      360      568      768
176      368      576      776
184      376      584      784
192      384      592      792
200      392      600      800
                           808
                           816
                           824
                           832
                           840
                           848
                           856
                           864
                           872
```

Question-02:

To find the numbers with 6 in range 1-1000 we used a “For Loop” with a built in function str(i) has been implemented.

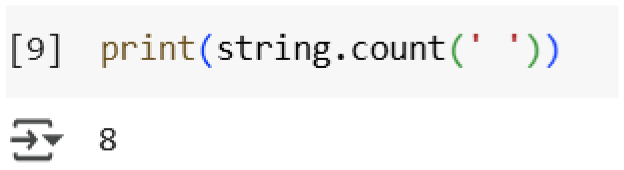


```
[8] num_with_6=[i for i in range(1,1001) if '6' in str(i)]
    print(num_with_6)
```

```
[6, 16, 26, 36, 46, 56, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 76, 86, 96, 106, 116, 126, 136, 146, 156, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 176, 186, 196, 206, 216, 22
236, 246, 256, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 276, 286, 296, 306, 316, 326, 336, 346, 356, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 376, 386, 396, 406, 416
426, 436, 446, 456, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 476, 486, 496, 506, 516, 526, 536, 546, 556, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 576, 586, 596, 600
601, 602, 603, 604, 605, 606, 607, 608, 609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637
638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674,
675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 706, 716, 726, 736, 746, 756, 760, 761, 762, 763, 764, 765,
766, 767, 768, 769, 776, 786, 796, 806, 816, 826, 836, 846, 856, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 876, 886, 896, 906, 916, 926, 936, 946, 956, 960, 961, 962, 963,
964, 965, 966, 967, 968, 969, 976, 986, 996]
```

Question-03:

To count the spaces in string built in function string.count(‘ ‘) has been implemented.

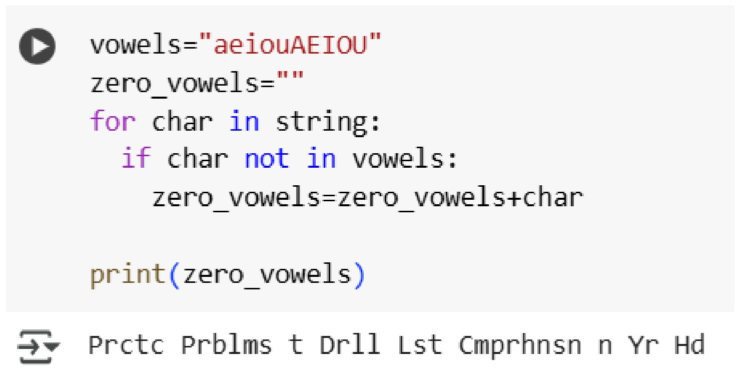


```
[9] print(string.count(' '))
```

```
8
```

Question-04:

To remove the vowels from the string & getting a new zero_vowel string we use “For loop” with “IF” condition is used.



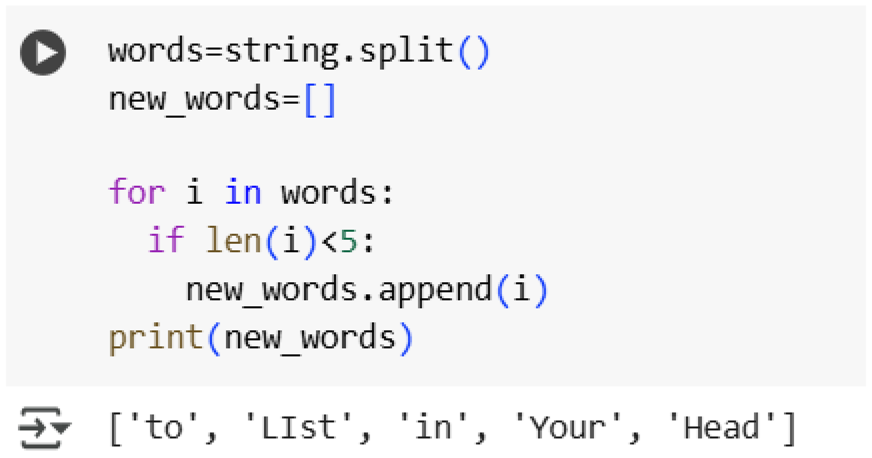
```python
vowels="aeiouAEIOU"
zero_vowels=""
for char in string:
  if char not in vowels:
    zero_vowels=zero_vowels+char

print(zero_vowels)
```

```
Prctc Prblms t Drll Lst Cmprhnsn n Yr Hd
```

Question-05:

To find words in string with letters less than 5 “If” condition with built in function len(i) in “For loop” has been used.



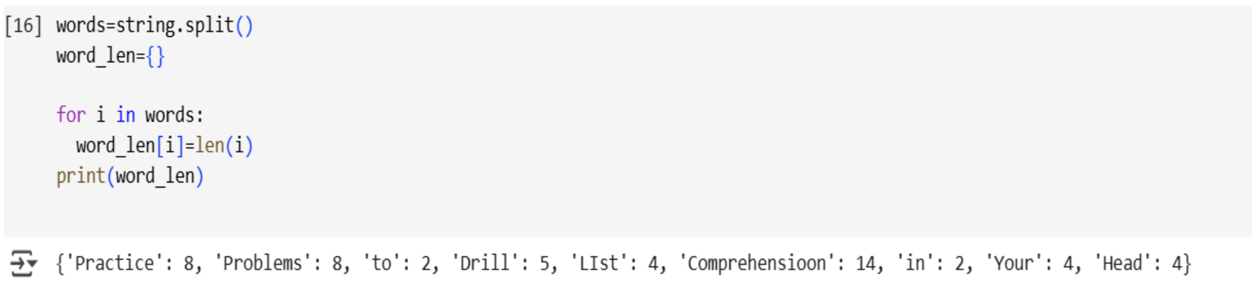
```python
words=string.split()
new_words=[]

for i in words:
  if len(i)<5:
    new_words.append(i)
print(new_words)
```

```
['to', 'LIst', 'in', 'Your', 'Head']
```

Question-06:

To count the length of each word in a string we used a built in function string.split() to split each word of the string & to count the letters in each word built in function len(i) has been used.



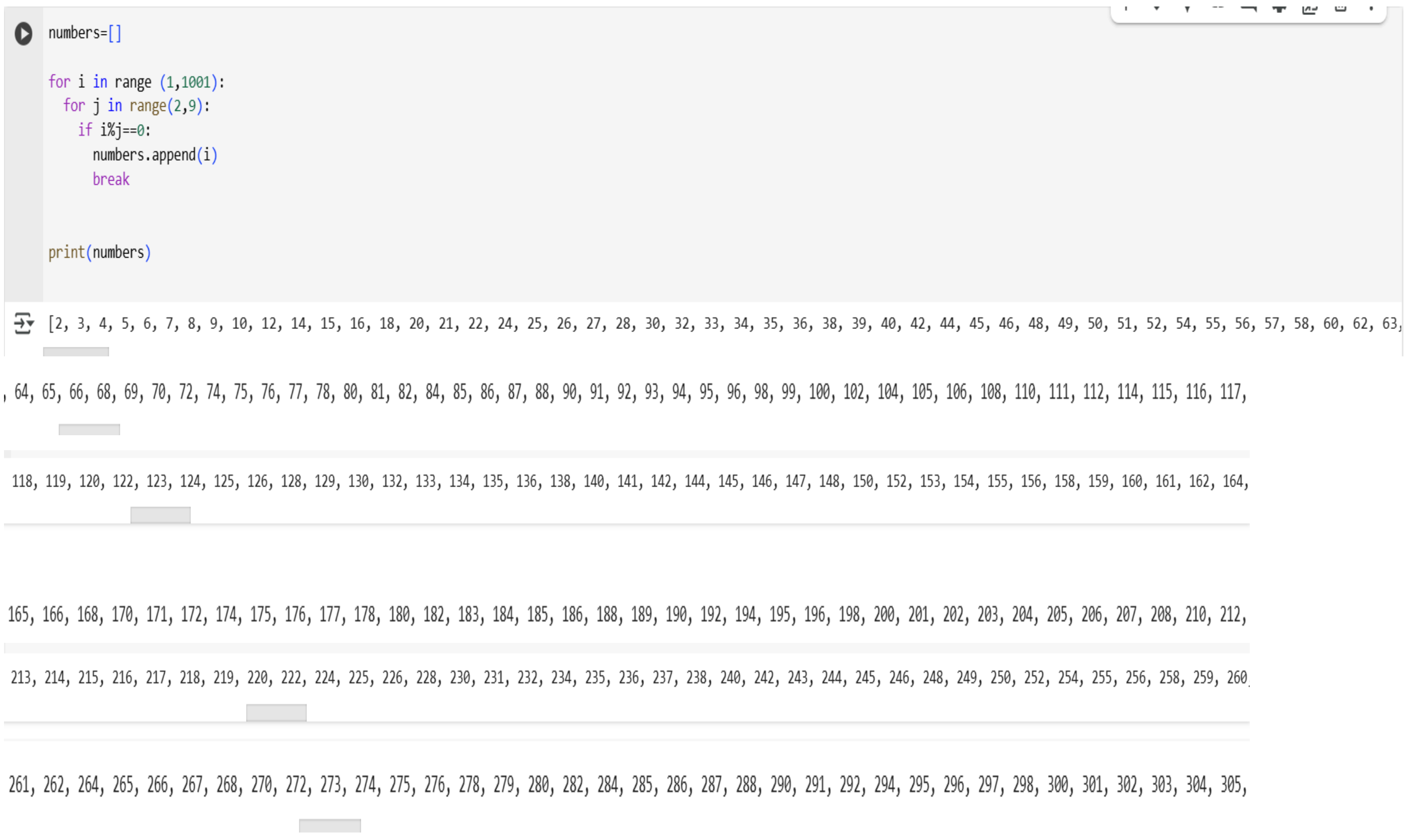
```
[16] words=string.split()
     word_len={}

     for i in words:
       word_len[i]=len(i)
     print(word_len)
```

```
{'Practice': 8, 'Problems': 8, 'to': 2, 'Drill': 5, 'LIst': 4, 'Comprehensioon': 14, 'in': 2, 'Your': 4, 'Head': 4}
```

Question-07:

To find all of the numbers from 1–1000 that are divisible by any single digit besides 1 (2–9) we used 2 loops with a built-in function append.

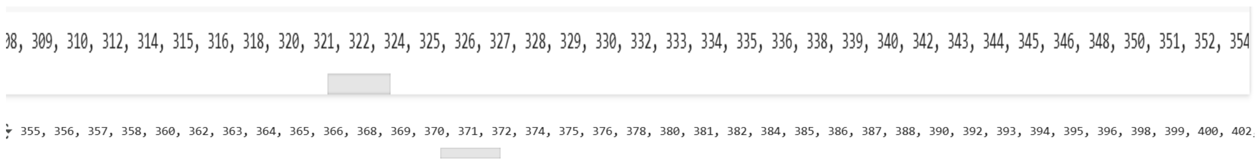


```
numbers=[]

for i in range (1,1001):
  for j in range(2,9):
    if i%j==0:
      numbers.append(i)
      break


print(numbers)
```

```
[2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14, 15, 16, 18, 20, 21, 22, 24, 25, 26, 27, 28, 30, 32, 33, 34, 35, 36, 38, 39, 40, 42, 44, 45, 46, 48, 49, 50, 51, 52, 54, 55, 56, 57, 58, 60, 62, 63,
, 64, 65, 66, 68, 69, 70, 72, 74, 75, 76, 77, 78, 80, 81, 82, 84, 85, 86, 87, 88, 90, 91, 92, 93, 94, 95, 96, 98, 99, 100, 102, 104, 105, 106, 108, 110, 111, 112, 114, 115, 116, 117,
118, 119, 120, 122, 123, 124, 125, 126, 128, 129, 130, 132, 133, 134, 135, 136, 138, 140, 141, 142, 144, 145, 146, 147, 148, 150, 152, 153, 154, 155, 156, 158, 159, 160, 161, 162, 164,
165, 166, 168, 170, 171, 172, 174, 175, 176, 177, 178, 180, 182, 183, 184, 185, 186, 188, 189, 190, 192, 194, 195, 196, 198, 200, 201, 202, 203, 204, 205, 206, 207, 208, 210, 212,
213, 214, 215, 216, 217, 218, 219, 220, 222, 224, 225, 226, 228, 230, 231, 232, 234, 235, 236, 237, 238, 240, 242, 243, 244, 245, 246, 248, 249, 250, 252, 254, 255, 256, 258, 259, 260
261, 262, 264, 265, 266, 267, 268, 270, 272, 273, 274, 275, 276, 278, 279, 280, 282, 284, 285, 286, 287, 288, 290, 291, 292, 294, 295, 296, 297, 298, 300, 301, 302, 303, 304, 305,
```



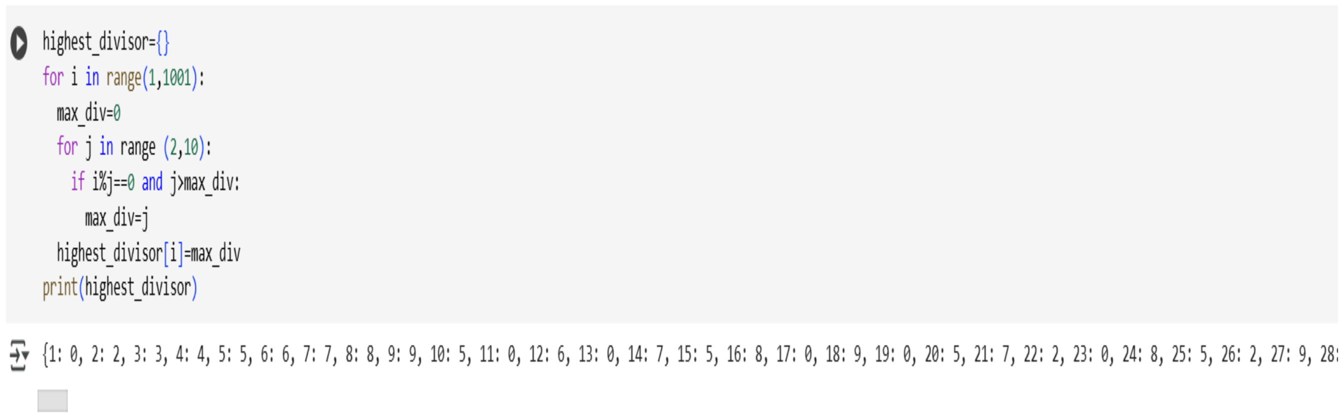
```
)8, 309, 310, 312, 314, 315, 316, 318, 320, 321, 322, 324, 325, 326, 327, 328, 329, 330, 332, 333, 334, 335, 336, 338, 339, 340, 342, 343, 344, 345, 346, 348, 350, 351, 352, 354
```

```
355, 356, 357, 358, 360, 362, 363, 364, 365, 366, 368, 369, 370, 371, 372, 374, 375, 376, 378, 380, 381, 382, 384, 385, 386, 387, 388, 390, 392, 393, 394, 395, 396, 398, 399, 400, 402,
```

(more outputs available)

Question-08:

To find the highest single digit any of the numbers in range 1-1000 is divisible by we used 2 "For Loops" & 2 initial variables to track the numbers in loops.



```python
highest_divisor={}
for i in range(1,1001):
  max_div=0
  for j in range (2,10):
    if i%j==0 and j>max_div:
      max_div=j
  highest_divisor[i]=max_div
print(highest_divisor)
```

```
{1: 0, 2: 2, 3: 3, 4: 4, 5: 5, 6: 6, 7: 7, 8: 8, 9: 9, 10: 5, 11: 0, 12: 6, 13: 0, 14: 7, 15: 5, 16: 8, 17: 0, 18: 9, 19: 0, 20: 5, 21: 7, 22: 2, 23: 0, 24: 8, 25: 5, 26: 2, 27: 9, 28:
```

(more outputs available)