In [4]:
```
#Find all of the numbers from 1-1000 that are divisible by 8
def divisibleBy8(number):
    return number%8==0

for i in range(1,1001):
    if(divisibleBy8(i)):
        print("I=",i,"")
```

```
I= 8
I= 16
I= 24
I= 32
I= 40
I= 48
I= 56
I= 64
I= 72
I= 80
I= 88
I= 96
I= 104
I= 112
I= 120
I= 128
I= 136
I= 144
I= 152
I= 160
I= 168
I= 176
I= 184
I= 192
I= 200
I= 208
I= 216
I= 224
I= 232
I= 240
I= 248
I= 256
I= 264
I= 272
I= 280
I= 288
I= 296
I= 304
I= 312
I= 320
I= 328
I= 336
I= 344
I= 352
I= 360
I= 368
I= 376
I= 384
I= 392
I= 400
I= 408
I= 416
I= 424
I= 432
I= 440
I= 448
I= 456
```

```
I= 464
I= 472
I= 480
I= 488
I= 496
I= 504
I= 512
I= 520
I= 528
I= 536
I= 544
I= 552
I= 560
I= 568
I= 576
I= 584
I= 592
I= 600
I= 608
I= 616
I= 624
I= 632
I= 640
I= 648
I= 656
I= 664
I= 672
I= 680
I= 688
I= 696
I= 704
I= 712
I= 720
I= 728
I= 736
I= 744
I= 752
I= 760
I= 768
I= 776
I= 784
I= 792
I= 800
I= 808
I= 816
I= 824
I= 832
I= 840
I= 848
I= 856
I= 864
I= 872
I= 880
I= 888
I= 896
I= 904
I= 912
```

```
I= 920
I= 928
I= 936
I= 944
I= 952
I= 960
I= 968
I= 976
I= 984
I= 992
I= 1000
```

In [ ]:
```
#Find all of the numbers from 1-1000 that have a 6 in them
```

In [11]:
```
def find6(lst):
    if(lst%10==6):
        return print(lst)
    elif(lst%100==6):
        return print(lst)
    elif(lst%1000==6):
        return print(lst)
    else:
        return 0


lst=[y for y in range(1,1000)]
result=list(map(find6,lst))
```

```
6
16
26
36
46
56
66
76
86
96
106
116
126
136
146
156
166
176
186
196
206
216
226
236
246
256
266
276
286
296
306
316
326
336
346
356
366
376
386
396
406
416
426
436
446
456
466
476
486
496
506
516
526
536
546
556
566
```

```
576
586
596
606
616
626
636
646
656
666
676
686
696
706
716
726
736
746
756
766
776
786
796
806
816
826
836
846
856
866
876
886
896
906
916
926
936
946
956
966
976
986
996
```

In [8]:
```
three = [n for n in range(0,1000) if '6' in str(n)]
print(three)
```

```
[6, 16, 26, 36, 46, 56, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 76, 86, 96, 1
06, 116, 126, 136, 146, 156, 160, 161, 162, 163, 164, 165, 166, 167, 168, 16
9, 176, 186, 196, 206, 216, 226, 236, 246, 256, 260, 261, 262, 263, 264, 265,
266, 267, 268, 269, 276, 286, 296, 306, 316, 326, 336, 346, 356, 360, 361, 36
2, 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, 51
6, 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, 61
3, 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, 64
4, 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, 67
5, 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, 76
0, 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, 89
6, 906, 916, 926, 936, 946, 956, 960, 961, 962, 963, 964, 965, 966, 967, 968,
969, 976, 986, 996]
```

In [13]:
```
#Count the number of spaces in a string (use string above)
def CheckNumOfSpace(string):
    c=0
    for i in range(0,len(string)):
        if string[i]==" ":
            c=c+1
    return c
string = "Practice Problems to Drill List Comprehension in Your Head."
print("The number of spaces in a string")
print(CheckNumOfSpace(string))
```

```
The number of spaces in a string
8
```

In [21]:
```
#Remove all of the vowels in a string (use string above)
def remove_vowels(string1):
    v=['a','e','i','o','u','A','E','I','O','U']
    for i in string1:
        if i not in v:
            string1=string1+i
    print(string1)

string1 = "Practice Problems to Drill List Comprehension in Your Head."
print(remove_vowels(string1))
```

```
Practice Problems to Drill List Comprehension in Your Head.Prctc Prblms t Drl
l Lst Cmprhnsn n Yr Hd.
None
```

In [23]:
```
#5. Find all of the words in a string that are less than 5 letters (use string
above)
def less_than_5words(string):
    word=string.split(" ")
    st=[]
    for i in word:
        if len(i)<5:
            st.append(i)
    return st

string = "Practice Problems to Drill List Comprehension in Your Head."
print(less_than_5words(string))
```

```
['to', 'List', 'in', 'Your']
```

In [30]:
```
#6.Use a dictionary comprehension to count the length of each word in a senten
ce (use string above)
def length_size(string):
    size=[]
    SplitWord=string.split()
    for i in SplitWord:
        size.append(len(i))
    print(size)
    print(SplitWord)
string = "Practice Problems to Drill List Comprehension in Your Head."
length_size(string)
```

```
[8, 8, 2, 5, 4, 13, 2, 4, 5]
['Practice', 'Problems', 'to', 'Drill', 'List', 'Comprehension', 'in', 'You
r', 'Head.']
```

In [31]:
```
#07Use a nested list comprehension to find all of the numbers from 1-1000 that
are divisible by any single digit besides 1 (2-9)
list = [ i for i in range(1, 1001) if any( i%j == 0 for j in range(2, 10))]
print(list)
```

```
[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, 5
4, 55, 56, 57, 58, 60, 62, 63, 64, 65, 66, 68, 69, 70, 72, 74, 75, 76, 77, 7
8, 80, 81, 82, 84, 85, 86, 87, 88, 90, 91, 92, 93, 94, 95, 96, 98, 99, 100, 1
02, 104, 105, 106, 108, 110, 111, 112, 114, 115, 116, 117, 118, 119, 120, 12
2, 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, 16
1, 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, 20
3, 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, 24
3, 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, 28
4, 285, 286, 287, 288, 290, 291, 292, 294, 295, 296, 297, 298, 300, 301, 302,
303, 304, 305, 306, 308, 309, 310, 312, 314, 315, 316, 318, 320, 321, 322, 32
4, 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, 36
3, 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, 40
4, 405, 406, 408, 410, 411, 412, 413, 414, 415, 416, 417, 418, 420, 422, 423,
424, 425, 426, 427, 428, 429, 430, 432, 434, 435, 436, 438, 440, 441, 442, 44
4, 445, 446, 447, 448, 450, 452, 453, 454, 455, 456, 458, 459, 460, 462, 464,
465, 466, 468, 469, 470, 471, 472, 474, 475, 476, 477, 478, 480, 482, 483, 48
4, 485, 486, 488, 489, 490, 492, 494, 495, 496, 497, 498, 500, 501, 502, 504,
505, 506, 507, 508, 510, 511, 512, 513, 514, 515, 516, 518, 519, 520, 522, 52
4, 525, 526, 528, 530, 531, 532, 534, 535, 536, 537, 538, 539, 540, 542, 543,
544, 545, 546, 548, 549, 550, 552, 553, 554, 555, 556, 558, 560, 561, 562, 56
4, 565, 566, 567, 568, 570, 572, 573, 574, 575, 576, 578, 579, 580, 581, 582,
584, 585, 586, 588, 590, 591, 592, 594, 595, 596, 597, 598, 600, 602, 603, 60
4, 605, 606, 608, 609, 610, 612, 614, 615, 616, 618, 620, 621, 622, 623, 624,
625, 626, 627, 628, 630, 632, 633, 634, 635, 636, 637, 638, 639, 640, 642, 64
4, 645, 646, 648, 650, 651, 652, 654, 655, 656, 657, 658, 660, 662, 663, 664,
665, 666, 668, 669, 670, 672, 674, 675, 676, 678, 679, 680, 681, 682, 684, 68
5, 686, 687, 688, 690, 692, 693, 694, 695, 696, 698, 699, 700, 702, 704, 705,
706, 707, 708, 710, 711, 712, 714, 715, 716, 717, 718, 720, 721, 722, 723, 72
4, 725, 726, 728, 729, 730, 732, 734, 735, 736, 738, 740, 741, 742, 744, 745,
746, 747, 748, 749, 750, 752, 753, 754, 755, 756, 758, 759, 760, 762, 763, 76
4, 765, 766, 768, 770, 771, 772, 774, 775, 776, 777, 778, 780, 782, 783, 784,
785, 786, 788, 789, 790, 791, 792, 794, 795, 796, 798, 800, 801, 802, 804, 80
5, 806, 807, 808, 810, 812, 813, 814, 815, 816, 818, 819, 820, 822, 824, 825,
826, 828, 830, 831, 832, 833, 834, 835, 836, 837, 838, 840, 842, 843, 844, 84
5, 846, 847, 848, 849, 850, 852, 854, 855, 856, 858, 860, 861, 862, 864, 865,
866, 867, 868, 870, 872, 873, 874, 875, 876, 878, 879, 880, 882, 884, 885, 88
6, 888, 889, 890, 891, 892, 894, 895, 896, 897, 898, 900, 902, 903, 904, 905,
906, 908, 909, 910, 912, 914, 915, 916, 917, 918, 920, 921, 922, 924, 925, 92
6, 927, 928, 930, 931, 932, 933, 934, 935, 936, 938, 939, 940, 942, 944, 945,
946, 948, 950, 951, 952, 954, 955, 956, 957, 958, 959, 960, 962, 963, 964, 96
5, 966, 968, 969, 970, 972, 973, 974, 975, 976, 978, 980, 981, 982, 984, 985,
986, 987, 988, 990, 992, 993, 994, 995, 996, 998, 999, 1000]
```

In [35]:
```
#08 For all the numbers 1-1000, use a nested list/dictionary comprehension to
 find the highest single digit any of the numbers is divisible by
def findMax(nums):
    for num in nums:
        maximum = 0
        for i in range(1, 10):
            if (num % i) == 0:
                maximum = max(i, maximum)
        if(maximum != 0):
            print(f'index: {num}, Maximum: {maximum}')

nums = [i for i in range(1,1001)]
findMax(nums)
```

```
index: 1, Maximum: 1
index: 2, Maximum: 2
index: 3, Maximum: 3
index: 4, Maximum: 4
index: 5, Maximum: 5
index: 6, Maximum: 6
index: 7, Maximum: 7
index: 8, Maximum: 8
index: 9, Maximum: 9
index: 10, Maximum: 5
index: 11, Maximum: 1
index: 12, Maximum: 6
index: 13, Maximum: 1
index: 14, Maximum: 7
index: 15, Maximum: 5
index: 16, Maximum: 8
index: 17, Maximum: 1
index: 18, Maximum: 9
index: 19, Maximum: 1
index: 20, Maximum: 5
index: 21, Maximum: 7
index: 22, Maximum: 2
index: 23, Maximum: 1
index: 24, Maximum: 8
index: 25, Maximum: 5
index: 26, Maximum: 2
index: 27, Maximum: 9
index: 28, Maximum: 7
index: 29, Maximum: 1
index: 30, Maximum: 6
index: 31, Maximum: 1
index: 32, Maximum: 8
index: 33, Maximum: 3
index: 34, Maximum: 2
index: 35, Maximum: 7
index: 36, Maximum: 9
index: 37, Maximum: 1
index: 38, Maximum: 2
index: 39, Maximum: 3
index: 40, Maximum: 8
index: 41, Maximum: 1
index: 42, Maximum: 7
index: 43, Maximum: 1
index: 44, Maximum: 4
index: 45, Maximum: 9
index: 46, Maximum: 2
index: 47, Maximum: 1
index: 48, Maximum: 8
index: 49, Maximum: 7
index: 50, Maximum: 5
index: 51, Maximum: 3
index: 52, Maximum: 4
index: 53, Maximum: 1
index: 54, Maximum: 9
index: 55, Maximum: 5
index: 56, Maximum: 8
index: 57, Maximum: 3
```

```
index: 58, Maximum: 2
index: 59, Maximum: 1
index: 60, Maximum: 6
index: 61, Maximum: 1
index: 62, Maximum: 2
index: 63, Maximum: 9
index: 64, Maximum: 8
index: 65, Maximum: 5
index: 66, Maximum: 6
index: 67, Maximum: 1
index: 68, Maximum: 4
index: 69, Maximum: 3
index: 70, Maximum: 7
index: 71, Maximum: 1
index: 72, Maximum: 9
index: 73, Maximum: 1
index: 74, Maximum: 2
index: 75, Maximum: 5
index: 76, Maximum: 4
index: 77, Maximum: 7
index: 78, Maximum: 6
index: 79, Maximum: 1
index: 80, Maximum: 8
index: 81, Maximum: 9
index: 82, Maximum: 2
index: 83, Maximum: 1
index: 84, Maximum: 7
index: 85, Maximum: 5
index: 86, Maximum: 2
index: 87, Maximum: 3
index: 88, Maximum: 8
index: 89, Maximum: 1
index: 90, Maximum: 9
index: 91, Maximum: 7
index: 92, Maximum: 4
index: 93, Maximum: 3
index: 94, Maximum: 2
index: 95, Maximum: 5
index: 96, Maximum: 8
index: 97, Maximum: 1
index: 98, Maximum: 7
index: 99, Maximum: 9
index: 100, Maximum: 5
index: 101, Maximum: 1
index: 102, Maximum: 6
index: 103, Maximum: 1
index: 104, Maximum: 8
index: 105, Maximum: 7
index: 106, Maximum: 2
index: 107, Maximum: 1
index: 108, Maximum: 9
index: 109, Maximum: 1
index: 110, Maximum: 5
index: 111, Maximum: 3
index: 112, Maximum: 8
index: 113, Maximum: 1
index: 114, Maximum: 6
```

```
index: 115, Maximum: 5
index: 116, Maximum: 4
index: 117, Maximum: 9
index: 118, Maximum: 2
index: 119, Maximum: 7
index: 120, Maximum: 8
index: 121, Maximum: 1
index: 122, Maximum: 2
index: 123, Maximum: 3
index: 124, Maximum: 4
index: 125, Maximum: 5
index: 126, Maximum: 9
index: 127, Maximum: 1
index: 128, Maximum: 8
index: 129, Maximum: 3
index: 130, Maximum: 5
index: 131, Maximum: 1
index: 132, Maximum: 6
index: 133, Maximum: 7
index: 134, Maximum: 2
index: 135, Maximum: 9
index: 136, Maximum: 8
index: 137, Maximum: 1
index: 138, Maximum: 6
index: 139, Maximum: 1
index: 140, Maximum: 7
index: 141, Maximum: 3
index: 142, Maximum: 2
index: 143, Maximum: 1
index: 144, Maximum: 9
index: 145, Maximum: 5
index: 146, Maximum: 2
index: 147, Maximum: 7
index: 148, Maximum: 4
index: 149, Maximum: 1
index: 150, Maximum: 6
index: 151, Maximum: 1
index: 152, Maximum: 8
index: 153, Maximum: 9
index: 154, Maximum: 7
index: 155, Maximum: 5
index: 156, Maximum: 6
index: 157, Maximum: 1
index: 158, Maximum: 2
index: 159, Maximum: 3
index: 160, Maximum: 8
index: 161, Maximum: 7
index: 162, Maximum: 9
index: 163, Maximum: 1
index: 164, Maximum: 4
index: 165, Maximum: 5
index: 166, Maximum: 2
index: 167, Maximum: 1
index: 168, Maximum: 8
index: 169, Maximum: 1
index: 170, Maximum: 5
index: 171, Maximum: 9
```

```
index: 172, Maximum: 4
index: 173, Maximum: 1
index: 174, Maximum: 6
index: 175, Maximum: 7
index: 176, Maximum: 8
index: 177, Maximum: 3
index: 178, Maximum: 2
index: 179, Maximum: 1
index: 180, Maximum: 9
index: 181, Maximum: 1
index: 182, Maximum: 7
index: 183, Maximum: 3
index: 184, Maximum: 8
index: 185, Maximum: 5
index: 186, Maximum: 6
index: 187, Maximum: 1
index: 188, Maximum: 4
index: 189, Maximum: 9
index: 190, Maximum: 5
index: 191, Maximum: 1
index: 192, Maximum: 8
index: 193, Maximum: 1
index: 194, Maximum: 2
index: 195, Maximum: 5
index: 196, Maximum: 7
index: 197, Maximum: 1
index: 198, Maximum: 9
index: 199, Maximum: 1
index: 200, Maximum: 8
index: 201, Maximum: 3
index: 202, Maximum: 2
index: 203, Maximum: 7
index: 204, Maximum: 6
index: 205, Maximum: 5
index: 206, Maximum: 2
index: 207, Maximum: 9
index: 208, Maximum: 8
index: 209, Maximum: 1
index: 210, Maximum: 7
index: 211, Maximum: 1
index: 212, Maximum: 4
index: 213, Maximum: 3
index: 214, Maximum: 2
index: 215, Maximum: 5
index: 216, Maximum: 9
index: 217, Maximum: 7
index: 218, Maximum: 2
index: 219, Maximum: 3
index: 220, Maximum: 5
index: 221, Maximum: 1
index: 222, Maximum: 6
index: 223, Maximum: 1
index: 224, Maximum: 8
index: 225, Maximum: 9
index: 226, Maximum: 2
index: 227, Maximum: 1
index: 228, Maximum: 6
```

```
index: 229, Maximum: 1
index: 230, Maximum: 5
index: 231, Maximum: 7
index: 232, Maximum: 8
index: 233, Maximum: 1
index: 234, Maximum: 9
index: 235, Maximum: 5
index: 236, Maximum: 4
index: 237, Maximum: 3
index: 238, Maximum: 7
index: 239, Maximum: 1
index: 240, Maximum: 8
index: 241, Maximum: 1
index: 242, Maximum: 2
index: 243, Maximum: 9
index: 244, Maximum: 4
index: 245, Maximum: 7
index: 246, Maximum: 6
index: 247, Maximum: 1
index: 248, Maximum: 8
index: 249, Maximum: 3
index: 250, Maximum: 5
index: 251, Maximum: 1
index: 252, Maximum: 9
index: 253, Maximum: 1
index: 254, Maximum: 2
index: 255, Maximum: 5
index: 256, Maximum: 8
index: 257, Maximum: 1
index: 258, Maximum: 6
index: 259, Maximum: 7
index: 260, Maximum: 5
index: 261, Maximum: 9
index: 262, Maximum: 2
index: 263, Maximum: 1
index: 264, Maximum: 8
index: 265, Maximum: 5
index: 266, Maximum: 7
index: 267, Maximum: 3
index: 268, Maximum: 4
index: 269, Maximum: 1
index: 270, Maximum: 9
index: 271, Maximum: 1
index: 272, Maximum: 8
index: 273, Maximum: 7
index: 274, Maximum: 2
index: 275, Maximum: 5
index: 276, Maximum: 6
index: 277, Maximum: 1
index: 278, Maximum: 2
index: 279, Maximum: 9
index: 280, Maximum: 8
index: 281, Maximum: 1
index: 282, Maximum: 6
index: 283, Maximum: 1
index: 284, Maximum: 4
index: 285, Maximum: 5
```

```
index: 286, Maximum: 2
index: 287, Maximum: 7
index: 288, Maximum: 9
index: 289, Maximum: 1
index: 290, Maximum: 5
index: 291, Maximum: 3
index: 292, Maximum: 4
index: 293, Maximum: 1
index: 294, Maximum: 7
index: 295, Maximum: 5
index: 296, Maximum: 8
index: 297, Maximum: 9
index: 298, Maximum: 2
index: 299, Maximum: 1
index: 300, Maximum: 6
index: 301, Maximum: 7
index: 302, Maximum: 2
index: 303, Maximum: 3
index: 304, Maximum: 8
index: 305, Maximum: 5
index: 306, Maximum: 9
index: 307, Maximum: 1
index: 308, Maximum: 7
index: 309, Maximum: 3
index: 310, Maximum: 5
index: 311, Maximum: 1
index: 312, Maximum: 8
index: 313, Maximum: 1
index: 314, Maximum: 2
index: 315, Maximum: 9
index: 316, Maximum: 4
index: 317, Maximum: 1
index: 318, Maximum: 6
index: 319, Maximum: 1
index: 320, Maximum: 8
index: 321, Maximum: 3
index: 322, Maximum: 7
index: 323, Maximum: 1
index: 324, Maximum: 9
index: 325, Maximum: 5
index: 326, Maximum: 2
index: 327, Maximum: 3
index: 328, Maximum: 8
index: 329, Maximum: 7
index: 330, Maximum: 6
index: 331, Maximum: 1
index: 332, Maximum: 4
index: 333, Maximum: 9
index: 334, Maximum: 2
index: 335, Maximum: 5
index: 336, Maximum: 8
index: 337, Maximum: 1
index: 338, Maximum: 2
index: 339, Maximum: 3
index: 340, Maximum: 5
index: 341, Maximum: 1
index: 342, Maximum: 9
```

```
index: 343, Maximum: 7
index: 344, Maximum: 8
index: 345, Maximum: 5
index: 346, Maximum: 2
index: 347, Maximum: 1
index: 348, Maximum: 6
index: 349, Maximum: 1
index: 350, Maximum: 7
index: 351, Maximum: 9
index: 352, Maximum: 8
index: 353, Maximum: 1
index: 354, Maximum: 6
index: 355, Maximum: 5
index: 356, Maximum: 4
index: 357, Maximum: 7
index: 358, Maximum: 2
index: 359, Maximum: 1
index: 360, Maximum: 9
index: 361, Maximum: 1
index: 362, Maximum: 2
index: 363, Maximum: 3
index: 364, Maximum: 7
index: 365, Maximum: 5
index: 366, Maximum: 6
index: 367, Maximum: 1
index: 368, Maximum: 8
index: 369, Maximum: 9
index: 370, Maximum: 5
index: 371, Maximum: 7
index: 372, Maximum: 6
index: 373, Maximum: 1
index: 374, Maximum: 2
index: 375, Maximum: 5
index: 376, Maximum: 8
index: 377, Maximum: 1
index: 378, Maximum: 9
index: 379, Maximum: 1
index: 380, Maximum: 5
index: 381, Maximum: 3
index: 382, Maximum: 2
index: 383, Maximum: 1
index: 384, Maximum: 8
index: 385, Maximum: 7
index: 386, Maximum: 2
index: 387, Maximum: 9
index: 388, Maximum: 4
index: 389, Maximum: 1
index: 390, Maximum: 6
index: 391, Maximum: 1
index: 392, Maximum: 8
index: 393, Maximum: 3
index: 394, Maximum: 2
index: 395, Maximum: 5
index: 396, Maximum: 9
index: 397, Maximum: 1
index: 398, Maximum: 2
index: 399, Maximum: 7
```

```
index: 400, Maximum: 8
index: 401, Maximum: 1
index: 402, Maximum: 6
index: 403, Maximum: 1
index: 404, Maximum: 4
index: 405, Maximum: 9
index: 406, Maximum: 7
index: 407, Maximum: 1
index: 408, Maximum: 8
index: 409, Maximum: 1
index: 410, Maximum: 5
index: 411, Maximum: 3
index: 412, Maximum: 4
index: 413, Maximum: 7
index: 414, Maximum: 9
index: 415, Maximum: 5
index: 416, Maximum: 8
index: 417, Maximum: 3
index: 418, Maximum: 2
index: 419, Maximum: 1
index: 420, Maximum: 7
index: 421, Maximum: 1
index: 422, Maximum: 2
index: 423, Maximum: 9
index: 424, Maximum: 8
index: 425, Maximum: 5
index: 426, Maximum: 6
index: 427, Maximum: 7
index: 428, Maximum: 4
index: 429, Maximum: 3
index: 430, Maximum: 5
index: 431, Maximum: 1
index: 432, Maximum: 9
index: 433, Maximum: 1
index: 434, Maximum: 7
index: 435, Maximum: 5
index: 436, Maximum: 4
index: 437, Maximum: 1
index: 438, Maximum: 6
index: 439, Maximum: 1
index: 440, Maximum: 8
index: 441, Maximum: 9
index: 442, Maximum: 2
index: 443, Maximum: 1
index: 444, Maximum: 6
index: 445, Maximum: 5
index: 446, Maximum: 2
index: 447, Maximum: 3
index: 448, Maximum: 8
index: 449, Maximum: 1
index: 450, Maximum: 9
index: 451, Maximum: 1
index: 452, Maximum: 4
index: 453, Maximum: 3
index: 454, Maximum: 2
index: 455, Maximum: 7
index: 456, Maximum: 8
```

```
index: 457, Maximum: 1
index: 458, Maximum: 2
index: 459, Maximum: 9
index: 460, Maximum: 5
index: 461, Maximum: 1
index: 462, Maximum: 7
index: 463, Maximum: 1
index: 464, Maximum: 8
index: 465, Maximum: 5
index: 466, Maximum: 2
index: 467, Maximum: 1
index: 468, Maximum: 9
index: 469, Maximum: 7
index: 470, Maximum: 5
index: 471, Maximum: 3
index: 472, Maximum: 8
index: 473, Maximum: 1
index: 474, Maximum: 6
index: 475, Maximum: 5
index: 476, Maximum: 7
index: 477, Maximum: 9
index: 478, Maximum: 2
index: 479, Maximum: 1
index: 480, Maximum: 8
index: 481, Maximum: 1
index: 482, Maximum: 2
index: 483, Maximum: 7
index: 484, Maximum: 4
index: 485, Maximum: 5
index: 486, Maximum: 9
index: 487, Maximum: 1
index: 488, Maximum: 8
index: 489, Maximum: 3
index: 490, Maximum: 7
index: 491, Maximum: 1
index: 492, Maximum: 6
index: 493, Maximum: 1
index: 494, Maximum: 2
index: 495, Maximum: 9
index: 496, Maximum: 8
index: 497, Maximum: 7
index: 498, Maximum: 6
index: 499, Maximum: 1
index: 500, Maximum: 5
index: 501, Maximum: 3
index: 502, Maximum: 2
index: 503, Maximum: 1
index: 504, Maximum: 9
index: 505, Maximum: 5
index: 506, Maximum: 2
index: 507, Maximum: 3
index: 508, Maximum: 4
index: 509, Maximum: 1
index: 510, Maximum: 6
index: 511, Maximum: 7
index: 512, Maximum: 8
index: 513, Maximum: 9
```

```
index: 514, Maximum: 2
index: 515, Maximum: 5
index: 516, Maximum: 6
index: 517, Maximum: 1
index: 518, Maximum: 7
index: 519, Maximum: 3
index: 520, Maximum: 8
index: 521, Maximum: 1
index: 522, Maximum: 9
index: 523, Maximum: 1
index: 524, Maximum: 4
index: 525, Maximum: 7
index: 526, Maximum: 2
index: 527, Maximum: 1
index: 528, Maximum: 8
index: 529, Maximum: 1
index: 530, Maximum: 5
index: 531, Maximum: 9
index: 532, Maximum: 7
index: 533, Maximum: 1
index: 534, Maximum: 6
index: 535, Maximum: 5
index: 536, Maximum: 8
index: 537, Maximum: 3
index: 538, Maximum: 2
index: 539, Maximum: 7
index: 540, Maximum: 9
index: 541, Maximum: 1
index: 542, Maximum: 2
index: 543, Maximum: 3
index: 544, Maximum: 8
index: 545, Maximum: 5
index: 546, Maximum: 7
index: 547, Maximum: 1
index: 548, Maximum: 4
index: 549, Maximum: 9
index: 550, Maximum: 5
index: 551, Maximum: 1
index: 552, Maximum: 8
index: 553, Maximum: 7
index: 554, Maximum: 2
index: 555, Maximum: 5
index: 556, Maximum: 4
index: 557, Maximum: 1
index: 558, Maximum: 9
index: 559, Maximum: 1
index: 560, Maximum: 8
index: 561, Maximum: 3
index: 562, Maximum: 2
index: 563, Maximum: 1
index: 564, Maximum: 6
index: 565, Maximum: 5
index: 566, Maximum: 2
index: 567, Maximum: 9
index: 568, Maximum: 8
index: 569, Maximum: 1
index: 570, Maximum: 6
```

```
index: 571, Maximum: 1
index: 572, Maximum: 4
index: 573, Maximum: 3
index: 574, Maximum: 7
index: 575, Maximum: 5
index: 576, Maximum: 9
index: 577, Maximum: 1
index: 578, Maximum: 2
index: 579, Maximum: 3
index: 580, Maximum: 5
index: 581, Maximum: 7
index: 582, Maximum: 6
index: 583, Maximum: 1
index: 584, Maximum: 8
index: 585, Maximum: 9
index: 586, Maximum: 2
index: 587, Maximum: 1
index: 588, Maximum: 7
index: 589, Maximum: 1
index: 590, Maximum: 5
index: 591, Maximum: 3
index: 592, Maximum: 8
index: 593, Maximum: 1
index: 594, Maximum: 9
index: 595, Maximum: 7
index: 596, Maximum: 4
index: 597, Maximum: 3
index: 598, Maximum: 2
index: 599, Maximum: 1
index: 600, Maximum: 8
index: 601, Maximum: 1
index: 602, Maximum: 7
index: 603, Maximum: 9
index: 604, Maximum: 4
index: 605, Maximum: 5
index: 606, Maximum: 6
index: 607, Maximum: 1
index: 608, Maximum: 8
index: 609, Maximum: 7
index: 610, Maximum: 5
index: 611, Maximum: 1
index: 612, Maximum: 9
index: 613, Maximum: 1
index: 614, Maximum: 2
index: 615, Maximum: 5
index: 616, Maximum: 8
index: 617, Maximum: 1
index: 618, Maximum: 6
index: 619, Maximum: 1
index: 620, Maximum: 5
index: 621, Maximum: 9
index: 622, Maximum: 2
index: 623, Maximum: 7
index: 624, Maximum: 8
index: 625, Maximum: 5
index: 626, Maximum: 2
index: 627, Maximum: 3
```

```
index: 628, Maximum: 4
index: 629, Maximum: 1
index: 630, Maximum: 9
index: 631, Maximum: 1
index: 632, Maximum: 8
index: 633, Maximum: 3
index: 634, Maximum: 2
index: 635, Maximum: 5
index: 636, Maximum: 6
index: 637, Maximum: 7
index: 638, Maximum: 2
index: 639, Maximum: 9
index: 640, Maximum: 8
index: 641, Maximum: 1
index: 642, Maximum: 6
index: 643, Maximum: 1
index: 644, Maximum: 7
index: 645, Maximum: 5
index: 646, Maximum: 2
index: 647, Maximum: 1
index: 648, Maximum: 9
index: 649, Maximum: 1
index: 650, Maximum: 5
index: 651, Maximum: 7
index: 652, Maximum: 4
index: 653, Maximum: 1
index: 654, Maximum: 6
index: 655, Maximum: 5
index: 656, Maximum: 8
index: 657, Maximum: 9
index: 658, Maximum: 7
index: 659, Maximum: 1
index: 660, Maximum: 6
index: 661, Maximum: 1
index: 662, Maximum: 2
index: 663, Maximum: 3
index: 664, Maximum: 8
index: 665, Maximum: 7
index: 666, Maximum: 9
index: 667, Maximum: 1
index: 668, Maximum: 4
index: 669, Maximum: 3
index: 670, Maximum: 5
index: 671, Maximum: 1
index: 672, Maximum: 8
index: 673, Maximum: 1
index: 674, Maximum: 2
index: 675, Maximum: 9
index: 676, Maximum: 4
index: 677, Maximum: 1
index: 678, Maximum: 6
index: 679, Maximum: 7
index: 680, Maximum: 8
index: 681, Maximum: 3
index: 682, Maximum: 2
index: 683, Maximum: 1
index: 684, Maximum: 9
```

```
index: 685, Maximum: 5
index: 686, Maximum: 7
index: 687, Maximum: 3
index: 688, Maximum: 8
index: 689, Maximum: 1
index: 690, Maximum: 6
index: 691, Maximum: 1
index: 692, Maximum: 4
index: 693, Maximum: 9
index: 694, Maximum: 2
index: 695, Maximum: 5
index: 696, Maximum: 8
index: 697, Maximum: 1
index: 698, Maximum: 2
index: 699, Maximum: 3
index: 700, Maximum: 7
index: 701, Maximum: 1
index: 702, Maximum: 9
index: 703, Maximum: 1
index: 704, Maximum: 8
index: 705, Maximum: 5
index: 706, Maximum: 2
index: 707, Maximum: 7
index: 708, Maximum: 6
index: 709, Maximum: 1
index: 710, Maximum: 5
index: 711, Maximum: 9
index: 712, Maximum: 8
index: 713, Maximum: 1
index: 714, Maximum: 7
index: 715, Maximum: 5
index: 716, Maximum: 4
index: 717, Maximum: 3
index: 718, Maximum: 2
index: 719, Maximum: 1
index: 720, Maximum: 9
index: 721, Maximum: 7
index: 722, Maximum: 2
index: 723, Maximum: 3
index: 724, Maximum: 4
index: 725, Maximum: 5
index: 726, Maximum: 6
index: 727, Maximum: 1
index: 728, Maximum: 8
index: 729, Maximum: 9
index: 730, Maximum: 5
index: 731, Maximum: 1
index: 732, Maximum: 6
index: 733, Maximum: 1
index: 734, Maximum: 2
index: 735, Maximum: 7
index: 736, Maximum: 8
index: 737, Maximum: 1
index: 738, Maximum: 9
index: 739, Maximum: 1
index: 740, Maximum: 5
index: 741, Maximum: 3
```

```
index: 742, Maximum: 7
index: 743, Maximum: 1
index: 744, Maximum: 8
index: 745, Maximum: 5
index: 746, Maximum: 2
index: 747, Maximum: 9
index: 748, Maximum: 4
index: 749, Maximum: 7
index: 750, Maximum: 6
index: 751, Maximum: 1
index: 752, Maximum: 8
index: 753, Maximum: 3
index: 754, Maximum: 2
index: 755, Maximum: 5
index: 756, Maximum: 9
index: 757, Maximum: 1
index: 758, Maximum: 2
index: 759, Maximum: 3
index: 760, Maximum: 8
index: 761, Maximum: 1
index: 762, Maximum: 6
index: 763, Maximum: 7
index: 764, Maximum: 4
index: 765, Maximum: 9
index: 766, Maximum: 2
index: 767, Maximum: 1
index: 768, Maximum: 8
index: 769, Maximum: 1
index: 770, Maximum: 7
index: 771, Maximum: 3
index: 772, Maximum: 4
index: 773, Maximum: 1
index: 774, Maximum: 9
index: 775, Maximum: 5
index: 776, Maximum: 8
index: 777, Maximum: 7
index: 778, Maximum: 2
index: 779, Maximum: 1
index: 780, Maximum: 6
index: 781, Maximum: 1
index: 782, Maximum: 2
index: 783, Maximum: 9
index: 784, Maximum: 8
index: 785, Maximum: 5
index: 786, Maximum: 6
index: 787, Maximum: 1
index: 788, Maximum: 4
index: 789, Maximum: 3
index: 790, Maximum: 5
index: 791, Maximum: 7
index: 792, Maximum: 9
index: 793, Maximum: 1
index: 794, Maximum: 2
index: 795, Maximum: 5
index: 796, Maximum: 4
index: 797, Maximum: 1
index: 798, Maximum: 7
```

```
index: 799, Maximum: 1
index: 800, Maximum: 8
index: 801, Maximum: 9
index: 802, Maximum: 2
index: 803, Maximum: 1
index: 804, Maximum: 6
index: 805, Maximum: 7
index: 806, Maximum: 2
index: 807, Maximum: 3
index: 808, Maximum: 8
index: 809, Maximum: 1
index: 810, Maximum: 9
index: 811, Maximum: 1
index: 812, Maximum: 7
index: 813, Maximum: 3
index: 814, Maximum: 2
index: 815, Maximum: 5
index: 816, Maximum: 8
index: 817, Maximum: 1
index: 818, Maximum: 2
index: 819, Maximum: 9
index: 820, Maximum: 5
index: 821, Maximum: 1
index: 822, Maximum: 6
index: 823, Maximum: 1
index: 824, Maximum: 8
index: 825, Maximum: 5
index: 826, Maximum: 7
index: 827, Maximum: 1
index: 828, Maximum: 9
index: 829, Maximum: 1
index: 830, Maximum: 5
index: 831, Maximum: 3
index: 832, Maximum: 8
index: 833, Maximum: 7
index: 834, Maximum: 6
index: 835, Maximum: 5
index: 836, Maximum: 4
index: 837, Maximum: 9
index: 838, Maximum: 2
index: 839, Maximum: 1
index: 840, Maximum: 8
index: 841, Maximum: 1
index: 842, Maximum: 2
index: 843, Maximum: 3
index: 844, Maximum: 4
index: 845, Maximum: 5
index: 846, Maximum: 9
index: 847, Maximum: 7
index: 848, Maximum: 8
index: 849, Maximum: 3
index: 850, Maximum: 5
index: 851, Maximum: 1
index: 852, Maximum: 6
index: 853, Maximum: 1
index: 854, Maximum: 7
index: 855, Maximum: 9
```

```
index: 856, Maximum: 8
index: 857, Maximum: 1
index: 858, Maximum: 6
index: 859, Maximum: 1
index: 860, Maximum: 5
index: 861, Maximum: 7
index: 862, Maximum: 2
index: 863, Maximum: 1
index: 864, Maximum: 9
index: 865, Maximum: 5
index: 866, Maximum: 2
index: 867, Maximum: 3
index: 868, Maximum: 7
index: 869, Maximum: 1
index: 870, Maximum: 6
index: 871, Maximum: 1
index: 872, Maximum: 8
index: 873, Maximum: 9
index: 874, Maximum: 2
index: 875, Maximum: 7
index: 876, Maximum: 6
index: 877, Maximum: 1
index: 878, Maximum: 2
index: 879, Maximum: 3
index: 880, Maximum: 8
index: 881, Maximum: 1
index: 882, Maximum: 9
index: 883, Maximum: 1
index: 884, Maximum: 4
index: 885, Maximum: 5
index: 886, Maximum: 2
index: 887, Maximum: 1
index: 888, Maximum: 8
index: 889, Maximum: 7
index: 890, Maximum: 5
index: 891, Maximum: 9
index: 892, Maximum: 4
index: 893, Maximum: 1
index: 894, Maximum: 6
index: 895, Maximum: 5
index: 896, Maximum: 8
index: 897, Maximum: 3
index: 898, Maximum: 2
index: 899, Maximum: 1
index: 900, Maximum: 9
index: 901, Maximum: 1
index: 902, Maximum: 2
index: 903, Maximum: 7
index: 904, Maximum: 8
index: 905, Maximum: 5
index: 906, Maximum: 6
index: 907, Maximum: 1
index: 908, Maximum: 4
index: 909, Maximum: 9
index: 910, Maximum: 7
index: 911, Maximum: 1
index: 912, Maximum: 8
```

```
index: 913, Maximum: 1
index: 914, Maximum: 2
index: 915, Maximum: 5
index: 916, Maximum: 4
index: 917, Maximum: 7
index: 918, Maximum: 9
index: 919, Maximum: 1
index: 920, Maximum: 8
index: 921, Maximum: 3
index: 922, Maximum: 2
index: 923, Maximum: 1
index: 924, Maximum: 7
index: 925, Maximum: 5
index: 926, Maximum: 2
index: 927, Maximum: 9
index: 928, Maximum: 8
index: 929, Maximum: 1
index: 930, Maximum: 6
index: 931, Maximum: 7
index: 932, Maximum: 4
index: 933, Maximum: 3
index: 934, Maximum: 2
index: 935, Maximum: 5
index: 936, Maximum: 9
index: 937, Maximum: 1
index: 938, Maximum: 7
index: 939, Maximum: 3
index: 940, Maximum: 5
index: 941, Maximum: 1
index: 942, Maximum: 6
index: 943, Maximum: 1
index: 944, Maximum: 8
index: 945, Maximum: 9
index: 946, Maximum: 2
index: 947, Maximum: 1
index: 948, Maximum: 6
index: 949, Maximum: 1
index: 950, Maximum: 5
index: 951, Maximum: 3
index: 952, Maximum: 8
index: 953, Maximum: 1
index: 954, Maximum: 9
index: 955, Maximum: 5
index: 956, Maximum: 4
index: 957, Maximum: 3
index: 958, Maximum: 2
index: 959, Maximum: 7
index: 960, Maximum: 8
index: 961, Maximum: 1
index: 962, Maximum: 2
index: 963, Maximum: 9
index: 964, Maximum: 4
index: 965, Maximum: 5
index: 966, Maximum: 7
index: 967, Maximum: 1
index: 968, Maximum: 8
index: 969, Maximum: 3
```

```
index: 970, Maximum: 5
index: 971, Maximum: 1
index: 972, Maximum: 9
index: 973, Maximum: 7
index: 974, Maximum: 2
index: 975, Maximum: 5
index: 976, Maximum: 8
index: 977, Maximum: 1
index: 978, Maximum: 6
index: 979, Maximum: 1
index: 980, Maximum: 7
index: 981, Maximum: 9
index: 982, Maximum: 2
index: 983, Maximum: 1
index: 984, Maximum: 8
index: 985, Maximum: 5
index: 986, Maximum: 2
index: 987, Maximum: 7
index: 988, Maximum: 4
index: 989, Maximum: 1
index: 990, Maximum: 9
index: 991, Maximum: 1
index: 992, Maximum: 8
index: 993, Maximum: 3
index: 994, Maximum: 7
index: 995, Maximum: 5
index: 996, Maximum: 6
index: 997, Maximum: 1
index: 998, Maximum: 2
index: 999, Maximum: 9
index: 1000, Maximum: 8
```

In [ ]: