1. Given the string s = 'Columbus', evaluate the following:
   1. s[0]
   2. s[0:1]
   3. s[1:]
   4. s[-1]
   5. s[3: :-1]
   6. s[2: :2]
2. Given the list myList = [3, 2, -1, [4, 7], 5], evaluate the following:
   1. myList[0]
   2. myList[0:1]
   3. myList[-1]
   4. myList[3]
   5. myList[:2]
   6. myList[2: :2]
3. Evaluate the following:
   1. 'a' in 'backache'
   2. [1, 2, 3] + [[11, 13, 12][1]] + [22, 33, 44, 55][1:]
   3. [3 for x in range(6)]
   4. [2\*y + 1 for y in [1, 3, 5, 7]]
   5. [x for x in range(3, 10) if x % 2 == 0]
   6. [len(w) for w in ['Go', 'Terriers']]
4. Write a function count\_ones(s) that takes in a string s of '0's and '1's and returns the number of '1's in the input.
5. Write a function swap\_bits(s) that takes in a string s of '0's and '1's and returns a string in which each bit in s has been swapped/replaced with the other bit. For example, swap\_bits('101011') should return '010100'.
6. Write a function num\_divisors(n) that returns the number of integers from 1 to n(inclusive) that divide n evenly. For example, num\_divisors(42) should return 8, because 1, 2, 3, 6, 7, 14, 21, and 42 are all divisors of 42.
7. Use the above num\_divisors(n) function in order to write a functionmost\_divisors(lst) that takes in a list of integers lst and returns the integer from that list with the most divisors. For instance, most\_divisors([2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]) should return 12.
8. Write a function count\_transitions(s) that takes in a string s of '0's and '1's and returns the number of times there is a transition from a '0' to a '1' or vice-versa in that input string. For example, count\_transitions('1110110000') should return 3.
9. Write a function longest\_string(lst) that takes in a list of strings lst as input and returns the longest string from that list. For example, longest\_string(['short', 'longer', 'sesquipedalian']) should return 'sesquipedalian'.
10. Write a function cycle(s, n) that takes in a string s of '0's and '1's and an integer nand returns the string in which s has shifted its last character to the initial position ntimes. For example, cycle('1110110000', 2) should return '0011101100'.
11. What is printed by the following working Python program?
12. def dog(x):
13. print('in dog, x is', x)
14. y = cat(x - 1) + cat(x + 2)
15. print('in dog, y is', y)
16. return y
17. def cat(y):
18. print('in cat, y is', y)
19. x = rat(y \* 2) + 3
20. print('in cat, x is', x)
21. return x
22. def rat(x):
23. print('in rat, x is', x)
24. return 2 \* x
25. y = dog(3)
26. print('at this level, y is', y)
27. What is printed by the following working Python program?
28. def mystery(x):
29. print('x is', x)
30. if x < 1:
31. return 2
32. else:
33. p = 6 - mystery(x - 1)
34. print('p is', p)
35. return p
36. y = mystery(3)

print('y is', y)