Name: _____ CM: _____ Section: _____ Grade: _____ of 10

1. Show the output of these expressions:

     print(3 + 3) _____   **print ("3" + "3")** _____

     Why are the outputs different?

2. What is the output of the code shown to the right?

```
nums = []
for k in range(5):
    nums = nums + [k * 2]
print(nums)
```

3. Suppose that we modified the code in the preceding problem
    by replacing the **nums = []** line with **nums = 0** and
    dropping the **[]** surrounding **k * 2**, so that the code
    becomes like that shown to the right.

```
nums = 0
for k in range(5):
    nums = nums + k * 2
print(nums)
```

    a. What is the output of the modified code? _____

    b. The name (variable) **nums** is now badly chosen.
        What would be a better name for it?

4. What happens in problem 2 if we forget the **nums = []** line altogether? Be specific.

5. Suppose that we modified the code in the preceding
    problem yet again, so that it now looks like the code shown
    to the right.

```
nums = ""
for k in range(5):
    nums = nums + str(k * 2)
print(nums)
```

    a. What is the output of the modified code?

    b. What would go wrong if we omitted the **str** function call?

6. Suppose that   *seq_of_seqs*   is a sequence of sequences, for example,

    [[1, 2, 3], [4, 5], [6], [7, 8, 9], []]

    Write code that would print the **length** of each inner sequence, each on its own line (so the above example would print   **3   2   1   3   0**   but each on its own line).

7. Repeat the previous problem, but now looping BACKWARDS from the **last** element in *seq_of_seqs* to the **first** element (so the above example would print   **0   3   1   2   3** but each on its own line).

8. The function shown to the right is intended to return   **True**   if the given sequence of numbers contains a negative number, and **False**   otherwise. For example:

    has_negative([5, 3, -4, 8]) should return **True**

    has_negative([5, 3, 4, 8]) should return **False**

    a. What does **has_negative**, as written, in fact return when the argument is **[5, 3, -4, 8]**?

    b. Mark up the code to indicate the changes needed to make the code correct.

```
def has_negative(numbers):
    for k in range(len(numbers)):
        if numbers[k] < 0:
            return True
        else:
            return False
```

9. The function shown to the right is intended to return **True**   if the given sequence of numbers is a *decreasing* sequence, that is, if each number in the sequence is less than or equal to the *next* number in the sequence. For example:

    is_decreasing([15, 11, 4, 4, 1]) should return **True**

    is_decreasing([15, 11, 4, 8, 1]) should return **False**   (since 8 is bigger than 4, its predecessor in the sequence).

    a. Fill in the blanks with **True** and **False**   in the appropriate places.

    b. The function has a small error in the FOR statement. Mark up the code to correct the error.

```
def is_decreasing(numbers):

    for k in range(len(numbers)):

        if numbers[k + 1] > numbers[k]:

            return _____

    return _____
```