

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

1. The following two functions both return the list `[1, 2, 3, ... n]`, for the given `n`. They are the same except for the bold-italicized lines.

<pre>def using_concatenation(n): new = [] for k in range(1, n + 1): new = new + [k] return new</pre>	<pre>def using_append(n): new = [] for k in range(1, n + 1): new.append(k) return new</pre>
---	--

With your instructor: open today's project and examine module `m0r_concatenation_vs_append`. Per the instructions in that module, read the code, run the module, and answer the questions in it (with your instructor's help as needed).

Then **circle** which of the above implementations *is better*. **Why is it better?**

2. Continuing the previous problem, circle **True** or **False** for each of the following.

Each time through the loop:

- The implementation on the **left** **** mutates **** `new`. True or False
 - The implementation on the **left** **** re-assigns **** `new`. True or False
 - The implementation on the **right** **** mutates **** `new`. True or False
 - The implementation on the **right** **** re-assigns **** `new`. True or False
3. Consider the code below. (The code is in two columns, but it is all one program.)

```
def increment_last_number(numbers):
    new = []
    for k in range(len(numbers)):
        new.append(numbers[k])
    new[len(new) - 1] = new[len(new) - 1] + 1
    return new

def main():
    r = [4, 20, 6, 10]
    s = increment_last_number(r)
    print(r)
    print(s)
```

When `main` runs, what does it print?

4. The function in the previous problem returned a new list that is a copy of the given list, except that the last number in the list is incremented by 1. Write the code for a `mutate_last_number` function that **mutates** its given list of numbers so that the last number in the list is incremented by 1. (Hint: it is a one-liner!)

```
def mutate_last_number(numbers):
```

- What advantage does `increment_last_number` have over `mutate_last_number`?
- What advantage does `mutate_last_number` have over `increment_last_number`?

7. Which of the following are patterns that the video presented for iterating through items in a sequence?

Check all that apply.

Beginning to end Selecting items Finding something
 Two places at once Parallel sequences Max or min

8. Complete the implementation of the following function in TWO ways:

<pre>def get_max(numbers): """ Returns the largest number in the given non-empty list. """</pre>	
<pre>biggest = numbers[0]</pre> <p><i>Put one solution below here:</i></p>	<pre>index = 0</pre> <p><i>The other solution below here:</i></p>

9. Suppose that instead of the largest number in the given non-empty list (as in the previous problem), you wanted to return the largest number **at an odd index (position)** in the given non-empty list. What change(s) would you make to the code in your answer to the previous problem?

10. Suppose that you wanted to find the largest **positive** number in a given non-empty list. That is a much harder problem than either of the preceding problems. Why?

11. What is the output of the following code?

```
def mystery(s):
    for k in range(1, len(s)):
        print(s[k - 1], s[k])

mystery("csse120")
```

12. Write one line of code to print both the first and last characters in the string variable called **clown**.

13. Write (to the right) a single line of code that has approximately the same effect as `nums = nums + [17]`, but **mutates** the `nums` list instead of re-assigning it.

14. Search online for “list remove python” to try to find the 3 functions/methods to remove an item from a list. List the names of those 3 functions/methods to the right. Then search for the Stack Overview post titled “Difference between ___, ___ and ___ on lists” (but replacing the underscores with the 3 names you found) and read its excellent explanation for the differences between the 3 functions/methods.