Write your name at the top of each page before you begin. [5 points]

1. [5 points] What does q1() print?

```python
def q1():
    count = 0
    for x in ['a', 'b', 'c']:
        for y in ['b', 'c', 'd']:
            if x == y:
                count += 1
    print(count)
```

2. [5 points] What does q2() print?

```python
def xcount(s):
    '''Sorry, no docstring to explain it'''
    count = 0
    for ch in s:
        if ch == "x":
            count += 1
    return count

def xavg(txt_list):
    nrows = len(txt_list)
    sum = 0
    for line in txt_list:
        sum += xcount(line)
    avg = sum / nrows
    return avg

def q2():
    print( xavg(["abxxcdxx", "xmx", "abcd"]))
```
3. [5 points] What does q3() print?

```python
def clamp(ar, floor):
    for i in range(len(ar)):
        if ar[i] < floor:
            ar[i] = floor

def q3():
    temps = [30, 31, 32, 34, 34, 34]
    clamp(temps, 32)
    ntemps = len(temps)
    sum = 0
    for tmp in temps:
        sum += tmp
    avgtemp = sum / ntemps
    print(avgtemp)
```

4. [5 points] What does q4() print?

```python
def periscope(x,y):
    x = 2 * x
    y = 2 * y
    return x - y

def q4():
    x = 7
    y = 5
    z = periscope(x,y)
    print(x + y + z)
```
5. [12 points] In our project flooding the cavern, we used a list of lists to represent a rectangular grid of cells. Here we test a list of lists to see if it could represent a rectangular grid. Finish the function is_rectangular, consistent with its docstring.

def is_rectangular(grid):
    """
    Check whether the rows of s are all the same length.
    Args:
        grid: A list of lists, e.g., [[1, 2, 3], [4, 5, 6]]
    Returns:
        True if each of the rows of grid have the same number of elements.
    Examples:
        is_rectangular([[1, 2], [3, 4], [5, 6]]) = True
        is_rectangular([[1, 2], [3, 4, 5], [6, 7]]) = False
        is_rectangular([[], []]) = True
        is_rectangular([[]]) = True
    """
    #Your code here
6. [13 points] Finish the function longest_dup below, consistent with its docstring.

```python
def longest_dup(s):
    """
    Measure the longest sequence of a single character in s.
    Args:
        s: A string containing only letters
    Returns:
        An integer, which is the length of the longest substring in
        s made up of all the same character.
    Examples:
        longest_dup("abcccdecl") = 3
        longest_dup("aaabbcccd") = 3
        longest_dup("aaaxaaxaaaxa") = 3
        longest_dup("abcde") = 1
        longest_dup("") = 0
    """
    # Your code here
```