Name: ______________________________ Recitation: ____ Andrew Id: ________________

15-112 Summer 2019 Quiz 1

Up to 50 minutes. No calculators, no notes, no books, no computers. Style will not be graded on quizzes. Show your work!

1. (5 points) **Syllabus Short Answer:** Where are TA office hours held?

2. (5 points) **Short Answer:** We discussed three types of errors in class: syntax, runtime, and logical errors. Suppose we have a block of code that has one of each error. Which error will Python report first and why?

3. (10 points) **Code Tracing:** Indicate what the following program prints. Place your answers (and nothing else) in the box under the code.

```
def ct1(s, n):
    for i in range(n):
        for j in range(n, -1, -1):
            if(j <= i):
                break
            num = int(s[i:j])
            print(num, end= " ")
            if(num % 2 == 1):
                print("odd num")
            elif(num % 5 == 2):
                print("cool num")
            elif(num**2 == 144):
                print("found it!")
                return num
            else:
                print()
    return num
print(ct1('112', 3))
```

4. (10 points) **Code Tracing:** Indicate what the following program prints. Place your answers (and nothing else) in the box under the code.

```
def ct2(s):
    res1 = ""
    res2 = ""
    count = 0
    for i in range(1,len(s)//2):
        res1 += s[1:len(s)-1:i]
        res2 += s[:(i*2)]
        print("res1 = %s len = %d" %(res1, len(res1)))
        print("res2 = %s len = %d" %(res2, len(res2)))

    for i in range(len(res1)):
        if(res1[i] == res2[i]):
            count += 1
    return count

print(ct2("qwertyui"))
```

5. (10 points) **Reasoning Over Code**: Find an argument for roc1 that makes it return True. Place your answer (and nothing else) in the box under the code.

```
def roc(s):
    t = "1a1b2c"
    assert(isinstance(s, str) and len(s) == len(t))
    length = len(s)
    result = ""
    for index in range(length):
        c1 = t[index]
        c2 = s[length - 1 - index]
        if(c1 == c2):
            result += c1
        elif(c1 in s):
            result = result.replace(c1, "")
    return result == "112"
```

6. (20 points) **Free Response:** Write the function longestRun(s, chars) that takes a possibly-empty string s and a second possibly-empty string of chars. We will say that a character is "good" if it is in the chars string (case insensitively, so "A" and "a" would match). The function should return the length of the longest consecutive run of good characters in the given string s. For example, consider: longestRun("abbcazBbcababb","bz"). This returns 3 (look for "zBb").

7. (40 points) **Free Response:** We will say that a positive integer is decreasing if the number is positive, and its digits strictly decrease from left to right. For example, 7654 is a decreasing number, while 2315 and 2333 are not. With this in mind, write the function nthDecreasingNumber(n), which returns the nth decreasing number, starting with zero. A couple test cases are shown below:

```
nthDecreasingNumber(0) = 1
nthDecreasingNumber(10) = 20
nthDecreasingNumber(46) = 91
nthDecreasingNumber(200) = 6520
```

Note: For full credit, you may not use strings in your solution. However, you will receive half credit for a fully correct solution that uses strings.