Name: ______________________ Recitation: ______ Andrew Id: ______________

15-112 Spring 2019 Quiz 3

Up to 20 minutes. No calculators, no notes, no books, no computers, **no extra paper.** Show your work!
Do not use list indexing, dictionaries, try/except, or recursion on this quiz.
You may import the math and/or string libraries and use their functions.
Note: We gave you lots of space, but that doesn't mean the functions need to be long!

1. (20 points) **Free Response:** Write the function `getAcronym(s)` that takes a string s and returns a string with only the first letters of each word. Assume s is composed of words separated by spaces. Case does not matter! For example:

   `getAcronym('Carnegie Mellon University')` returns `'CMU'`

   `getAcronym('Reasoning over Code')` returns `'RoC'`

   `getAcronym('abc def ghi 123')` returns `'adg1'`

QUIZ IS ON BOTH SIDES OF THE PAGE

2. (10 points) **Short Answer:** In the recitation video, Arman plans his solution to collapseWhiteSpace(s) by stepping through the example string character-by-character, removing repeat whitespace characters until he reaches the end of the string. Which of the three **problem solving strategies** from the algorithmic thinking notes **best** describes his approach?

3. (25 points) **Free Response:** Write the function `collapseWhitespace(s)` (which you saw in the recitation videos) that takes a string s and returns an equivalent string except that each sequential occurrence of whitespace in the string is replaced by a single space. You may assume s does not have leading or trailing whitespace. Note: You do not necessarily have to solve it exactly like the video, though you can if you like!

   Here are some examples:

   `collapseWhitespace('a\t\t\tb\n\nc')` returns `'a b c'`

   `collapseWhitespace('a      bc    \t    d')` returns `'a bc d'`

   `collapseWhitespace('a b c')` returns `'a b c'`

4. (10 points) **Short Answer:** Write a good comment of 10 words or less that could be used to **summarize** the function in the box below.

```
import string
def mystery(s):
    result=''
    for c in s:
        if c in string.ascii_lowercase:
            result+=c.upper()
        elif c in string.ascii_uppercase:
            result+=c.lower()
        else:
            result+=c
    return result
```

5. (20 points) **Code Tracing:** Indicate what the following program prints. Place your answer (and nothing else) in the box to the right of the code.

```
def ct(s):
    r = ''
    t = ''
    for i in range(len(s)):
        if (s[i] in s[i+1:]):
            n = s[i:]
            n = n.count(s[i])
            print(str(i)+str(n))
            r += s[i]
        else:
            t = s[i] + t
    return r + ',' + t

print(ct('xyzxyx'))
```

6. (15 points) **Reasoning Over Code**: Find an input value for `s` that makes `roc(s)` return `True`. Place your answer (and nothing else) in the box to the right of the code.

```
def roc(s):
    c1=s[0]

    for i in range(1,len(s)):
        c2=s[i]
        if ord(c1) != ord(c2) + 1:
            return False
        c1=c2

    assert(len(s)==4)
    assert('e' in s)

    return True
```