

Name: \_\_\_\_\_ Section: \_\_\_ Andrew Id: \_\_\_\_\_

**15-112 Spring 2016 Quiz 4xa**

**\* Up to 20 minutes. No calculators, no notes, no books, no computers.**

**\* No lists or recursion! \* To receive credit (in Code Tracing), show your work.**

**1. Short Answer** [5 pts]

What does ("a%%bc%de%6.1fg" % (123, 45.67)) evaluate to?

**2. Code Tracing** [25 pts]: Indicate what this prints. Place your answer (and nothing else) in the box below the code.

```
def ct1(s, p):
    i = 0
    result = ""
    for c in s:
        z = ord(p[i]) - ord('a')
        result += chr(ord(c) + z)
        while z > 0:
            result = result[1 : ] + result[0]
            z -= 1
        i = (i + 1) % len(p)
    return result
print(ct1("awesome", "abd"))
```

**3. Reasoning Over Code** [10 pts]: Find arguments for the following function that makes it return True. You only need one set of arguments, even if there are multiple correct answers.

```
def roc(s, t):
    assert((s != "" and t != "") and (s in t))
    result = ""
    for i in range(len(s)):
        if (i % 2) == 0: result += t[i]
        else: result += t[-1-i]
    return (result == s)
```

4. **Free Response: encodeAtbashCipher(s)** [60 pts]

The Atbash cipher encodes words by substituting the first letter of an alphabet with the last letter, the second letter with the second-to-last letter, and so on. For example, "a" becomes "z", "z" becomes "a", "b" becomes "y", and "y" becomes "b". With this in mind, write a function `encodeAtbashCipher(s)` that takes a possibly-empty string and returns that string with the Atbash cipher applied to every word with more than 4 characters. You may assume that the string is all lowercase, has no punctuation or digits or other non-lowercase characters (except spaces), and words are separated by a single space.

5. **Bonus/Optional Code Tracing** [5 pts]: Indicate what this prints. Circle your answer.

```
def bonus(s, t):
    while (t != ""): (s,t) = (s+str(ord(t[len(s)%len(t)]) - ord(t[-1])), t[1:-1])
    return eval(s)
print(bonus("", string.ascii_lowercase[5:12]))
```