

Name: _____ Andrew Id: _____

15-112 Fall 2023 Quiz 4

Up to 20 + 5 minutes (finish within 20 minutes for 1-point proficiency bonus)

No calculators, no notes, no books, no computers. Show your work!

Do not use lists, dictionaries, try/except, or recursion on this quiz.

1. **Code Tracing:** Indicate what the following program prints. Place your answer (and nothing else) in the box next to the code.

(a) (4 points) CT1

```
def ct1(s):  
    if s.find('bad'):  
        s.replace('bad', 'good')  
    print(s)  
    s = s[:-3] + 'fun'  
    t = ''  
    n = 0  
    for c in s:  
        if not c.isdigit():  
            t += c  
        else:  
            n += int(c)  
    print(n, t)  
    return s
```

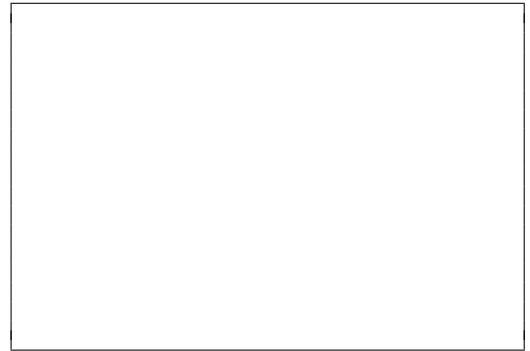
```
ct1('programming15is112bad')
```



(b) (4 points) CT2


```
def ct2(s, t):
    u = s.lower()
    print("u:",u)
    for c in s.lower():
        u += t[ord(c) - ord('a'):]
    return u

print(ct2('ACd', 'abc'))
```



2. (4 points) **Reasoning Over Code:** Find an argument, `s`, for the following function to cause it to return `True`. Place your answer (and nothing else) in the box below the code. Hint: Use triple single quotes `'''` or triple double quotes `"""` to define the value.

```
def roc(s):
    cnt = 0
    a = "0"
    b = ""
    for t in s[::-1].split('\n'):
        cnt = cnt + 1
        if len(t) != 2 or not t[0].isdigit():
            return False
        a += t[0]
        b += t[1]
    return cnt == 2 and a + b == "042ih"
```



3. (8 points) **Free Response: bestWord**

Write the function `bestWord(message, letterScores)` that takes in a message, a space-separated string of words, and returns the *"best"* word within the message as defined by `letterScores`. `letterScores` is an alphabetical lowercase string where each character occurs at most once, and a character's index in the string defines the score (starting at 1) of the corresponding character. The score of a word is the sum of all the scores of each character in that word (ignoring case). Return the best word that comes first in alphabetical order in case of ties.

For example, if `letterScores = "abc"`, then `"a"` has 1 point, `"b"` has 2 points, and `"c"` has 3 points. Any character not in `letterScores` should be awarded 0 points. So, using `letterScores = "abc"`, the word `"cat"` is worth 4 points (`"c"` has 3 points, `"a"` has 1 point, and `"t"` has 0 points).

For example, `bestWord("I love programming", "iaov")` should return `"love"` because `"love"` is the highest scoring word, with a score of $0+3+4+0 = 7$.

For example, `bestWord("Programming is FUN", "ufo")` should return `"FUN"`, because although `"FUN"` and `"Programming"` both have a highest score of 3, `"FUN"` is the word that comes first in alphabetical order.

Here are more examples:

```
assert(bestWord("one test", "eo") == "one") # the score of "one" is 2+0+1=3,
                                             # and the score of "test" is 0+1+0+0=1
assert(bestWord("two TESTS", "et") == "TESTS") # the score of "TESTS" is 2+1+0+2+0=5,
                                             # while "two" is 2+0+0=2
assert(bestWord("The Last Test", "xyz") == "Last") # All words have a score equal to zero,
                                                    # but "Last" comes first alphabetically
```

You may continue your work on the back of this page...

Additional Space for Answer to Question 3