

## Quiz 3 CT and ROC (50 pts)

Time Limit: 15 mins

Name: \_\_\_\_\_ AndrewID: \_\_\_\_\_

---

This part does not contain any free responses. Do not run any code when completing this part of Quiz 3.

**Time limit: 15 mins**



## Part 2 — Code Trace / Reasoning Over Code (50 pts total)

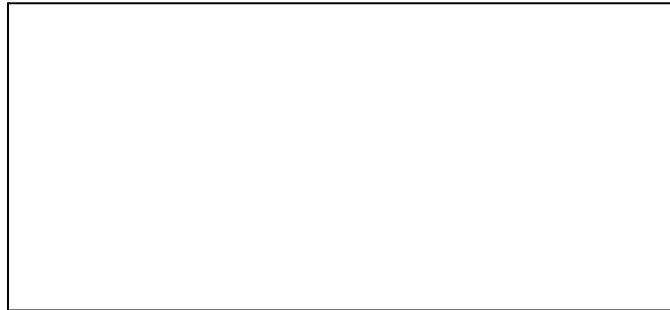
### Code Trace 1 (10 pts)

What does the following code print?

```
import string
def ct(s, t):
    n = min(len(s), len(t))
    res = ''
    for i in range(n):
        res += string.ascii_uppercase[ord(t[i]) - ord(s[i])]
    res += ',' * 3
    res += s[n:] + t[n:]
    return res

s = 'bY.9m'
t = 'eW.2mAXc'
print(ct(s, t))
```

**Output:**



### Code Trace 2 (10 pts)

What does the following code print?

```
def ct(L):  
    M = [v for v in L if type(v) == str]  
    N = [v for v in L if v not in M]  
    print(sorted(M))  
    print(sorted(N))  
    N += [max(M), min(M)]  
    return N  
  
print(ct(['a', 1, 'bc', 51, 1, 2, 'ba']))
```

**Output:**

A large, empty rectangular box with a black border, intended for the user to write the output of the code.

### Code Trace 3 (10 pts)

What does the following code print?

```
import copy
def ct(L):
    M = L
    N = copy.copy(L)
    print(N.append(51))
    print(N.pop(1))
    M.insert(2, 'xyz')
    L = L * 2
    L[3] = 100
    print(L)
    print(M)
    print(N)

L = [1, 5, 112]
ct(L)
print(L)
```

**Output:**

A large, empty rectangular box with a thin black border, intended for the user to write the output of the code trace.

### Reasoning Over Code 1 (10 pts)

Find an argument for the function `roc(s)` that makes it return `True`. Write your answer (and nothing else) in the box below.

```
def f(n):
    if n < 100:
        return False
    t = (n % 10) + (n // 10 % 10)
    while n > 0:
        p = (n % 10) + (n // 10 % 10)
        if p != t:
            return False
        n //= 100
    return True

def roc(s):
    c = 0
    d = 0
    for v in s.split('-'):
        c += 1
        if v.isdigit():
            if not f(int(v)):
                return False
        else:
            d += 1
    return (c == 4) and (d == 2)
```

## Reasoning Over Code 2 (10 pts)

Fill in the blanks so  $f(s) == g(s)$  for all strings  $s$ .

```
'''
def f(s):
    if not s.isalpha():
        return False
    n = 1
    t = ''
    for c1 in s:
        if c1 in t:
            continue
        t += c1
        m = 0
        for c2 in s:
            if c1 == c2:
                m += 1
        n *= m
    return n

def g(s):
    if not s.isalpha():
        return False
    n = 1
    for c in string.ascii_letters:
        if _____:
            n *= _____
    return n

'''
```

*This page intentionally left blank for scratch work*



*This page intentionally left blank for scratch work*

*This page intentionally left blank for scratch work*