Name: ____________________ Andrew Id: ____________

15-112 Spring 2022 Quiz 5

Up to 25 minutes. No calculators, no notes, no books, no computers. Show your work!
Do not use dictionaries, sets, try/except, or recursion on this quiz.

1. (8 points) **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 f(l, e):
    l += e
    return l

def ct1(s, l):
    r = ""
    for i in range(len(s)):
        if i not in l:
            r += s[i]
            print("A:", r)
        elif (i % 2 == 0):
            l = f(l, [i])
            print("B:", l)
        else:
            l = l + [i]
            print(f"C: {i}")
    return r

s="cmu"
L = [1,2]
ct1(s,L )
print(L)
```

(b) (4 points) CT2

```
def g(a):
    r = []
    for j in a:
        r.append(str(j))
    print(r)
    return r

def ct2(l):
    t=""
    x=g(l)
    for y in x:
        t += y
    print(t+t)
    print(t.replace("72","00"))
    d = int(t)
    d -= sum(l)
    print(d)

ct2([2,7])
```

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.

```python
def roc(s):
    assert(len(s) == 6)
    r = ""
    for c in s:
        if c.isdigit():
            n = int(c)
            if s.find(c) != n or n%2 != 0:
                return False
        else:
            r += c
    return r == "cmu"
```

3. (4 points) **Free Response (part I)**

   Do not use dictionaries, sets, try/except, or recursion on this problem. If you do, you will receive a 0.

   Write the function `isHandyWord(word, hand)` that takes two strings and returns `True` if the string `word` can be formed by some arrangement of some subset of the characters in `hand`, ignoring case (so "A" and "a" are the same), `False` otherwise.

   Both strings consists of **only alphabetical characters**.

```
assert( isHandyWord("Read", "adre") == True)
assert( isHandyWord("hello", "LLohe") == True)
assert( isHandyWord("hello", "Lohe") == False) # one L is missing
assert( isHandyWord("notes", "") == False)
assert( isHandyWord("in", "zz") == False)
assert( isHandyWord("lecture", "eetucrlabydf) == True)
assert( isHandyWord("a", "abcdef") == True)
```

   Write your answers on the following pages. Do not write your answer on this page.

4. (4 points) **Free Response (part II)**

   Do not use dictionaries, sets, try/except, or recursion on this problem. If you do, you will receive a 0.

   Implement the function `nondestructiveReplaceNonHandyWords(L, hand)` which takes a list of strings `L` and a string `hand` and returns a new string where all **non-handy** words occurring in `L` are replaced by the string consisting of a single dash character `-`, **non-destructively**. You may assume that your implementation of `isHandyWord(s, hand)` works, even if yours does not.

   Therefore, your function should behave like this:

```
L = ['Read','The', 'LecturE','NOTES']
print(nondestructiveReplaceNonHandyWords(L, "adehlnorstu"))
```

   Outputs:

```
['Read','The', '-', 'NOTES']
```

   In the example, note that `LecturE` was replaced by `-`.

   Write your answers on the following pages. Do not write your answer on this page.

Free Response answers:

Free Response answers: