

Name: _____ Andrew Id: _____

15-112 Fall 2021 Quiz 7

Up to 25 minutes. No calculators, no notes, no books, no computers. Show your work!

Do not use try/except 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.

```
def ct1(n,d):
    while n > 0:
        d[n%10] = d.get(n%10,n) + 1
        print(d)
        n //= 10

d={1:5,4:3}
print(ct1(112,d))
```

2. (4 points) **Reasoning Over Code:** Find an argument, L, for the function roc1 to cause it to return **True**. Place your answer (and nothing else) in the box below the code.

```
def f(L, d):
    if (len(L) == 0):
        return [ ]
    else:
        assert(d < L[0] and L[0] < 10)
        return [L[0]//d] + f(L[1:], d)

def roc1(L):
    return (f(L[1:], L[0]) == [1, 4])
```

3. (4 points) **Recursive Free Response**

Do not use loops, iteration, list/string functions that imply iteration (e.g., count, len) or try/except on this problem. If you do, you will receive a 0.

Implement the function `recursiveVowelCount(s)` which takes a string `s`, and returns the number of vowels in `s`. You can assume that `s` only contains lowercase characters and non-alphabetical characters. The vowels are "a", "e", "i", "o", and "u". So, for example:

```
assert(recursiveVowelCount("abc def a zyzyz") == 3) # two a's and one e
assert(recursiveVowelCount("hello") == 2) # one e and one o
assert(recursiveVowelCount("bcd") == 0) # no vowels
```

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

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

Write the function `commonKeysDictionary(L)` that takes a list of dictionaries and combines them into a single dictionary. The combined dictionary should contain only keys that are present in all the dictionaries in `L`. The value of each key in the combined dictionary should be the set of values associated with that key from the dictionaries in `L`. For example given: `L = [{"a": 1, "b": 2}, {"a": 3, "c": 4}, {"a": 5}]` `commonKeysDictionary(L)` returns `{"a": {1,3,5}}`, because the only key that is present in all the three dictionaries in `L` is `a`. More examples:

```
L = [{"a": 1, "b": 2, "c": 42}, {"a": 1, "c": 4}]
assert( commonKeysDictionary(L) == {"a": {1}, "c": {4,42}})
```

```
L = [{"a": 1, "b": 2}, {"a": 3, "c": 4}, {"a": 5}]
assert( commonKeysDictionary(L) == {"a": {1,3,5}})
```

```
L=[]
assert( commonKeysDictionary(L) == dict())
```