

15-112 Spring 2023 Quiz 8

Up to 20 + 10 minutes (finish within 20 minutes for 20% proficiency bonus). No calculators, no notes, no books, no computers. Show your work!

Do not use try/except, dictionaries, or sets on this quiz

1. (3 points) **Short Answer:** Consider the following code:

```
def f(a):  
    t = 0  
    for e in a:  
        if t in a:  
            t = t + 1  
    return t
```

Big-O time efficiency of the function if:

- (a) a is a `list` _____.
 - (b) a is a `set` _____.
 - (c) a is a `dict` _____.
2. (5 points) **Code Tracing:** Indicate what the following program prints. Place your answer (and nothing else) in the box next to the code. Ensure strings are enclosed with quotes, and the uppercase and lowercase are distinguishable.

```
def f(L, h):  
    for item in L:  
        if item in h:  
            h[item] += 2  
    print("b:",h)  
  
def ct1(L):  
    d = dict()  
    for i in range(len(L)):  
        d[sum(L[i:])] = L[i]  
    print("a:",d)  
    print(f(L, d))  
    for k in range(1,4,2):  
        if k in d:  
            del d[k]  
    return d  
print("c:",ct1([1,2,0,1]))
```

3. (4 points) **Reasoning Over Code:** Find an argument, `d`, for the following function to cause it to return `True`. Place your answer (and nothing else) in the box below the code. Make sure strings are enclosed with quotes.

```
def roc(d):  
    assert(isinstance(d, dict) and len(d) == 2)  
    s = set()  
    for k in d:  
        if not len(d[k]) == 2:  
            return False  
        s.add(k)  
        s.add(d[k])  
    return s == {"a", "b", "cd"}
```

4. (8 points) **Free Response:** 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 consist 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)
```

After you write your solution, indicate in the box below the efficiency of your algorithm using Big-O notation.
(2 points)

BONUS: +4 points if your solution runs in $O(N)$ time, where N is the sum of the length of the strings

Answer space for Question 4