

Name: \_\_\_\_\_ Andrew Id: \_\_\_\_\_

### 15-112 Spring 2023 Quiz 4

Up to 20 minutes. No calculators, no notes, no books, no computers. Show your work!

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

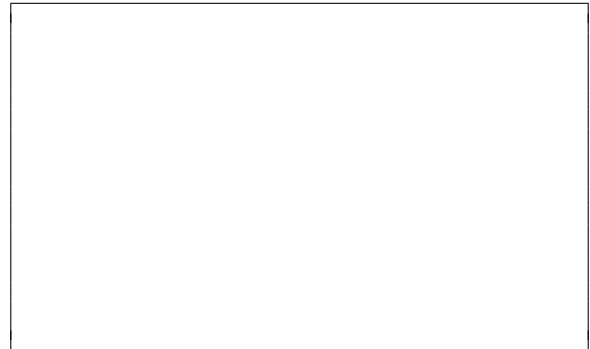
1. (8 points) **Code Tracing:** Indicate what the following program prints. Place your answer (and nothing else) in the box next to the code.

**Hint:** Draw a box and arrow diagram!

**Hint:** It prints 4 lines total.

```
def ct(a):  
    b = a  
    c = a.copy()  
    b[0] = 15  
    a.append("wow")  
    c[3] = c[3] // 2  
    b = b[:2] + ["hi"] + b[2:]  
    c.remove(2)  
    a[-2] = 112  
    print(a)  
    print(b)  
    print(c)
```

```
z = ["Feb", 2, "quiz", 4]  
ct(z)  
print(z)
```



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

```
def roc(L):  
    if (not isinstance(L, list)):  
        return False  
    A = []  
    B = []  
    while L != []:  
        A.extend([L[-1]])  
        B = [L[0]] + B  
        L = L[1:-1]  
    return A + B == [2,3,4,5]
```

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

Write the **destructive** function `removeRuns(L)` that takes a non-empty list of integers `L` and destructively modifies `L` by removing any values that would otherwise produce a run of consecutive equal values in `L`.

In other words, `L` is modified such that any sub-sequence of two or more occurrences of the same value ( $n, n, \dots$ ) is replaced with a single occurrence of the value ( $n$ ). As a result, no consecutive repeated values can be seen in `L` after calling the function.

**Note:** the final `L` can still have repeated values, but not consecutive.

For example...

```
L1 = [4,5,1,1,3,4]
```

```
removeRuns(L1) # substitutes the run 1,1 with 1
```

```
assert(L1 == [4,5,1,3,4])
```

```
L2 = [4,4,4,2,2,3,4]
```

```
removeRuns(L2) # substitutes the runs 4,4,4 with 4, and 2,2 with 2
```

```
assert(L2 == [4,2,3,4])
```

```
L3 = [1,1,1,1,1,1,1]
```

```
removeRuns(L3) # substitutes the run 1,1,1,1,1,1,1 with 1
```

```
assert(L3 == [1])
```

**Alternative:** You can solve the problem in a non-destructive way with a penalty of **-2 points**. If so, your function must return the result as a new list, and `L` should not be modified.

Tick below to solve it non-destructively with a penalty, and your work will be graded accordingly.

- I choose to solve the task using a non-destructive solution for a 2-point penalty.

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

**THIS PAGE IS INTENTIONALLY LEFT BLANK.  
NOTHING ON THIS PAGE WILL BE GRADED.**