

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

### 15-112 Fall 2023 Quiz 6

Up to 20 + 5 minutes (finish within 20 minutes for 1-point proficiency bonus)

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. Ensure strings are enclosed with quotes and the uppercase and lowercase are distinguishable.

```
def ct(a):
    b = a
    c = a[::-1]
    b[0] = 15
    a.append("wow")
    print(c)
    c[2] = c[2] // 2
    b = b[:2] + ["hi"] + b[2:]
    c.remove(2)
    a[-2] = 112
    print(a)
    print(b)
    print(c)

z = ["Oct", 5, "quiz", 6]
ct(z)
print(z)
```

2. (10 points) **Free Response:** Implement the function `foldTypeRun(L)`, which is designed to **destructively** modify a non-empty mixed-type list `L`. This list contains both single-digit integers and characters. The goal is to group consecutive equal-type values in `L` as follows: any run of consecutive integers should be combined into a single integer value whose digits correspond to the run, and all runs of consecutive characters should be transformed into a string with characters representing the run.

The function should modify `L` so that any subsequence with two or more occurrences of the same type is replaced with a single occurrence, following the abovementioned rules. Consequently, after calling the function, there should be no consecutive values of the same type (string or integer) in `L`.

Your function should be mutating/destructive. For full credit, **you are not allowed to empty the input list `L`**.

**Note:** the final `L` may still have repeated types, but they should not appear consecutively.

For example...

```
L = [4, 2]
foldTypeRun(L)
assert(L == [42])

L = ['b', 'y', 'e']
foldTypeRun(L)
assert(L == ['bye'])

L = [1, 2, 3, 'h', 'e', 'l', 'l', 'o', 1, 5, 1, 1, 2]
foldTypeRun(L)
assert(L == [123, 'hello', 15112]) # there were three runs with the same type
                                   # 1, 2, 3: become 123
                                   # 'h','e','l','l','o' become 'hello'
                                   # 1,5,1,1,2 become 15112

L = ['f', 'a', 'l', 'l', 2, 0, 2, 3, 'y', 'e', 'y']
foldTypeRun(L)
assert(L == ['fall', 2023, 'yey'])
```

**Alternative:** You can solve the problem in a non-destructive way with a penalty of **-4 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 4-point penalty.

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

Additional Space for Answer to Question 2