Name: _____ Andrew Id: _____

15-112 Spring 2026 Quiz 7

Up to 30 minutes. No calculators, no notes, no books, no computers. Show your work!
Do not use try/except on this quiz.

1. **Code Tracing**: Indicate what the following programs print. Place your answers (and nothing else) in the boxes below the code.

   (a) (4 points) CT1

```
def ct(s, i=0):
    if i >= len(s):
        return ""

    print(s[i], i)

    if i % 2 == 0:
        return s[i] + ct(s, i+3)
    else:
        return ct(s, i+1) + s[i]

print(ct("abcde"))
```

2. (7 points) **Free Response**: Remove Adjacent Duplicates

   Write the function `removeAdjDuplicates(s)` that takes a string `s` and returns a new string where consecutive duplicate characters are reduced to a single occurrence.

   Your solutions to this problem must be entirely recursive. No loops or iterative functions are allowed; their use will result in a zero score for this problem.

   Consider the following examples:

```
assert(removeAdjDuplicates("aaabbcaaa") == "abca")
assert(removeAdjDuplicates("a") == "a")
assert(removeAdjDuplicates("aaaa") == "a")
assert(removeAdjDuplicates("abbbcdd") == "abcd")
assert(removeAdjDuplicates("") == "")
```

3. (9 points) **Free Response**: Subset Sum

   Write the function `hasSubsetSum(L, target)` that takes a list of integers `L` and an integer `target`, and returns `True` if there exists any subset of elements in `L` whose sum is exactly equal to `target`. Otherwise, return `False`.

   Your solutions to this problem must be entirely recursive. No loops or iterative functions are allowed; their use will result in a zero score for this problem.

   Consider the following examples:

```
assert(hasSubsetSum([4, 1, 7, 9], 10) == True)   # 1 + 9
assert(hasSubsetSum([3, 1, 7, 9], 5) == False)
assert(hasSubsetSum([2, 4, 6], 6) == True)       # 6
assert(hasSubsetSum([2, 4, 6], 0) == True)       # empty subset
assert(hasSubsetSum([], 0) == True)
assert(hasSubsetSum([], 5) == False)
```