

Name: _____ Andrew Id: _____

15-112 Fall 2024 Quiz 9

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

1. (5 points) **Code Tracing:** Indicate what the following program prints. Place your answer (and nothing else) in the box below the code.

```
def ct(L, b):
    if b:
        print(f"{L}")
    if len(L) == 1 and L[0] % 2 == 1:
        return L[0]
    elif len(L) == 1:
        return 0

    x = len(L)//2
    return max(ct(L[:x], b), ct(L[x:], b))

print(ct([3, 16, 11, 20], True))
print(ct([6, 10, 21, 16, 24, 17, 7, 22, 5, 3, 14, 26, 15, 20, 2], False))
```

2. (5 points) **Free Response:** Join

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

Write the function `join(L, sep)` which, given a list of words `L` and a separator character `sep` joins all of the words from `L` into a single string where the words are separated by the `sep` character.

For example, consider the following testcases:

```
L = ["hi", "there", "my", "name", "is", "ahmed"]
assert join(L, " ") == "hi there my name is ahmed"
assert join(L, "1") == "hi1there1my1name1is1ahmed"
assert join([], "Q") == ""
```

3. (10 points) **Free Response:** Summing Pairs

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

Write the function `findSummingPairs(L, target)` that takes a list of integers `L` and an integer `target` that returns a list of all the pairs of integers in `L` that sum up to `target`. Each pair should be represented as a tuple `(a, b)`. The pairs should be included in the returned list in the same order that they appear in the original list.

For example, consider the following testcases:

```
L = [1, 3, 2, 2, 4, 5, -1, 3, 1]
print(findSummingPairs(L, 1)) # Prints [(2, -1), (2, -1)]
print(findSummingPairs(L, 5)) # Prints [(1, 4), (3, 2), (3, 2), (2, 3), (2, 3), (4, 1)]
print(findSummingPairs(L, 4)) # Prints [(1, 3), (1, 3), (3, 1), (2, 2), (5, -1), (3, 1)]
```

Hint: Our solution uses a recursive helper function.