# CS 152 Data Structures

Recursive Functions Worksheet

**Name:** ________________________________________

## Background

Recursive functions are functions that call themselves to solve a problem. A recursive function typically has:

- A **base case** that stops the recursion.
- A **recursive case** that reduces the problem and calls the function again.

Understanding recursive functions is a key skill for solving problems like traversing trees, generating permutations, and many others.

## Part 1: Predicting Output

**Problem 1.** What does the following function return when called with `mystery(4)`?

```python
def mystery(n):
    if n == 0:
        return 1
    else:
        return n * mystery(n - 1)
```

**Problem 2.** What does the following function return when called with `countdown(3)`?

```python
def countdown(n):
    if n == 0:
        return "Liftoff!"
    else:
        return str(n) + " " + countdown(n - 1)
```

## Part 2: Convert to Iteration

Rewrite the `mystery` function from Problem 1 using a `while` loop instead of recursion.

**Your Solution Below:**

## Part 3: Recursive Function Practice

Write a recursive function called `next_prime_after_double(n)` that takes an integer `n`, doubles it, and returns the next prime number that is greater than the doubled value.

**Steps:**

- Write a helper function `is_prime(num)` to check if a number is prime.
- Double the input `n`.
- Recursively find the next prime number greater than `2 * n`.

**Optional:** Try writing an iterative version of the same function.

*End of Worksheet*