Name: _____

# EMSE 4571 / 6571: Quiz 4 (R)

**Rules**:

- Work alone; no outside help of any kind is allowed.
- No calculators, no notes, no books, no computers, no phones.

## 1. Code Tracing: 40 points

Consider the following function:

```
f <- function(x) {
    x <- x + 1
    y <- 2*x
    cat(x, '\n')
    if ((x %% 2) == 0) {
        x <- x + 1
    } else if (x < 0) {
        y <- y %/% x
        cat(x)
        cat('wooo!')
    } else {
        y <- y %% x
    }
    cat(y)
}
```

After running the code below in R, write in the space below what will print in the console. Be careful to write any potential line breaks on new lines.

```
f(3)
```

Solution:

```
4
8
```

(Turn over for the remaining quiz questions)

## 2. Short Response: 20 points

Write one reason why you should always write a test function along with any function you write.

1. They help you understand the problem. 2. They verify that a function is working as expected.

## 3. Complete the test function: 40 points

Consider the following function:

```
isEven <- function(n) {
    return((n %% 2) == 0)
}
```

Complete the test function below, `test_isOdd()`, by writing three valid test cases. At least one of your tests should be for an outcome where `isOdd()` is expected to return `TRUE`, and another test should be for an outcome where `isOdd()` is expected to return `FALSE`

This is just an example:

```
test_isOdd <- function() {
    cat("Testing isOdd()...")
    # Write test cases here:

    # Test case 1:
    stopifnot(isOdd(2) == FALSE)

    # Test case 2:
    stopifnot(isOdd(3) == TRUE)

    # Test case 3:
    stopifnot(isOdd(-2) == FALSE)

    cat("Passed!\n")
}
```

## Bonus - TRUE or FALSE: 3 points

The trick of using `n %% 2 == 0` to determine if `n` is an even number works for both positive and negative numbers.

TRUE