1. (20 pts.) The following algorithm returns true if any two of the three numbers are equal. Add assert statements that describe what is true (or should be true) before the procedure starts, before each if statement, and at the end of the procedure.

```plaintext
procedure anyEqual(a, b, c)
    // need assertion here
    r := false
    // need assertion here
    if a = b then r := true
    // need assertion here
    if b = c then r := true
    // need assertion here
    if a = c then r := true
    // need assertion here
    return r
```

2. (20 pts.) The following algorithm returns the median of three numbers. Add assert statements that describe what is true (or should be true) before the procedure starts, before each if statement, and at the end of the procedure. You may simply assert “m is the median of a, b, and c” rather than trying to figure out a way to state this as a logical statement.

```plaintext
procedure median(a, b, c)
    // need assertion here
    x := a
    y := b
    // need assertion here
    if x > y then
        x := b
        y := a
    // need assertion here
    if y < c then m := y
    else if x > c then m := x
    else m := c
    // need assertion here
    return m
```
3. (20 pts.) Do Exercise 3.1.10. Use the book’s pseudocode. Add assert statements that
describe what is true (or should be true) before the procedure starts, before the loop,
at some point within the loop, and at the end of the procedure. That is, there should
be four assert statements.

4. (20 points) For each of the code segments in the following exercises from Section 3.3,
how many times will each statement will be executed? What is the total number of
statements executed?

3.3.2, 3.3.3

For loops should be translated into while loops for the purpose of counting statements.

5. (10 pts.) Prove that \( n^2 + 2n + 3 \) is \( O(n^2) \). Find values for \( C \) and \( k \) and prove that they
work.

6. (10 pts.) Prove that \( n^2 - 2n - 3 \) is \( \Omega(n^2) \). Find values for \( C \) and \( k \) and prove that they
work.