

## Lab Two

Identifying Equivalence partitions is largely an exercise in Logic. Here are two problems to practice that sort of thinking. Come up with the appropriate equivalence partitions and then invent test cases for the partitions.

Both of these examples involve “interdependent” variables—that is, the values of some variables determine the acceptable values of the others.

### Part One: Triangle Classifier

**Identify the partitions and test cases** for a function that takes in three integers (a, b, and c) as sides of a triangle, and determines whether the triangle is equilateral, isosceles, scalene, or invalid.

Remember to account for error hiding.

### Part Two: Date Validator

**Identify the partitions and test cases** for a function that takes in an int for month, an int for day, and an int for year, and determines whether the combinations are accurate. You can assume the boolean `isLeapYear(int year);` validator function is a separate unit that doesn’t require testing. That is, you don’t need to come up with partitions to test all the possible types of leap year, just make sure you account for leap years and non-leap years.

### Submission:

You can submit this in two different ways:

- 1) You could just come up with two tables listing the test, the test partitions, and the test cases (include both the inputs and the expected outputs in the test cases).
- 2) You could create a Git repo with the actual tests. I encourage you to try this—though we haven’t had much time to explore such frameworks yet.