

Lab One

Imagine a simple function that tests whether a given year of the Gregorian Calendar, entered as an integer, is a leap year. The function returns true if the year entered is a leap year, and false if it is not a leap year.

While taking an int as a parameter and returning a Boolean are the simplest requirements—let's assume those work.

Here are the additional requirements for the function:

1. If the integer entered is evenly divisible by 400, the function should return true.
2. Otherwise, if the integer is evenly divisible by 100, the function should return false.
3. Otherwise, if the integer is evenly divisible by 4, the function should return true.
4. Otherwise, the function should return false.

Part One: Write the Tests

Write a set of test cases (inputs and expected outputs) that you think would be **sufficient** to establish that this function is working properly.

These tests can be in actual code (with comments) or pseudo code.

Part Two: Analyze the Tests

In a table, list:

| Test Case | Inputs | Expected Outputs | What the test shows. |
|-------------|--------|------------------|--|
| Test case 1 | 2000 | true | Requirement 1: Divisible by 400 results in true. |

In a short paragraph, discuss the rationale for your decision that your test suite is sufficient to establish that the system works without failure.

Part Three: Devil's Advocate

Come up with a few ways that a system might pass your tests, but still fail. For example, the most obvious system would only send back accurate values for the inputs in the test cases. e.g.

```
if (year == 2000)  
    return true;
```