Assignment #1

This assignment is to be done individually or by teams of two students MAXIMUM (please read the policy about cheating). You are required to hand in a zip file containing your answers to the questions as well as your JUNIT implementation.

Problem 1

The date of Easter in a given year is March (22 + A + B), where A and B are determined as follows:

- A = the remainder of (19C + 24) / 30
- B = the remainder of (2D + 4E + 6A + 5) / 7
- C = the remainder of (year / 19)
- D = the remainder of (year / 4)
- E = the remainder of (year / 7)

This approach works for any year between 1584 and 4098 (inclusive). It computes an integer date value which is the day in March for Easter. However, if the result is one of the years 1954, 1981, 2049, or 2076, then the computed date value must be reduced by 7. And, if the date value is > 31, then the result is a day in April, so the date is calculated as date-31.

The easter program uses the class **EasterCalculator** method **easterDate** to calculate the easter date given a year. The header of the method is as follow:

```
public static MyDate easterDate(int year)
// returns a date corresponding to the easter day of
// the year given if  1583 < year < 4099
// returns null if not
```

Question 1.1 (15%)

Using the Equivalence Class Partitioning and the Boundary Value Analysis approaches, design black box tests for the method easterDate. Show:

- your equivalence classes with a short descriptive note
- the boundary values that should be checked.

Input Condition	*Valid Classes*	*Invalid Classes*	*Boundary Values*

Question 1.2 (10%)

Write enough test cases to cover all the equivalence classes, and boundary values identified in question 1.1.

Provide a table showing the link between your test data and the equivalence classes. This table should have the following format:

Test Case Number	Test Data	Expected results	Covers Equivalence	Boundary Values

You can find the dates of Easter at
https://tlarsen2.tripod.com/anthonypolumbo/apeasterdates.html

Question 1.3 (20%)

Implement your test suite using **JUnit** and the provided **EasterCalculatorTest**. You are required to hand the source code of your test suite.

Question 1.4 (5%)

Report your test results by providing a table with the following format (the test case numbers correspond to the ones in Question 1.2).

Test Case Number	Expected results	Actual results	Verdict (Pass/Fail)

Problem 2

The following describe a part of an elevator behavior.

The elevator has a set of buttons (one for each floor). These buttons illuminate when pressed and cause the elevator to stop at the corresponding floor.

Each floor, except the first and top floors, has two buttons, one to request an elevator traveling up (up-button) and one to request an elevator traveling down (down-button). The first floor only has an up-button and the top floor only has a down-button. These buttons illuminate when pressed. The illumination is canceled when the elevator stops at the floor traveling toward the desired direction.

For security reason, when the charge carried by the elevator is superior or equal to some threshold value, the elevator won't stop at a floor unless there is a request from within the elevator to stop at that floor.

The elevator always stops at the top floor when traveling up, and at the first floor when traveling down.

This problem concerns the elevator controller behavior when approaching a

floor.

The elevator has a sensor that is triggered as it nears each floor. The elevator controller must then decide whether or not to stop the elevator at that floor depending on the direction in which the elevator is traveling, the requests at the floor and the requests made from inside the elevator. The elevator controller also turns off illumination of the buttons that have to. For example:

- if the elevator is traveling *UP*,
- the floor up-button has been pressed,
- the corresponding elevator button for the floor has not been pressed, and,
- the charge carried by the elevator is less than the threshold value,

the elevator controller must stop the elevator and turn the floor up-button illumination off.

Your are asked to design test cases to verify the elevator behavior when approaching a floor using cause-effects graphing. Assume a single elevator.

Question 2.1 (5%)

Identify the *causes* and the *effects* for this problem.

Question 2.2 (20%)

Draw a ***cause-effect graph*** for this problem.

Question 2.3 (10%)

Provide a Boolean formula corresponding to each of the possible effects.

Question 2.5 (10%)

Provide a decision table obtained by applying *Each-Condition/All-Conditions* approach.

Question 2.5 (5%)

Give five (5) test cases derived from the decision table in question 2.4.