# Homework: Test Techniques

## Equivalence Partitioning / Boundary Value Analysis – Income Checker

Now that you are familiar with the Equivalence Partitioning / Boundary Value Analysis Techniques, let's recall [**The Income Checker App**](http://softuni-qa-loadbalancer-2137572849.eu-north-1.elb.amazonaws.com/income-checker/) from the QA Basics course, that categorizes the given **monthly income** into one of the following categories: "**low**", "**mid**", "**high"**. It works as follows:

* If the income is less than 1000, returns **"low"**
* If the income between 1000 (inclusively) and 3000 (exclusively), returns **"mid"**
* If the income is equal or bigger than 3000, returns **"high"**
* If the income is negative, returns **"error"**

**Your task is:**

**Equivalence Partitioning:** Divide the possible input values of the "**income**" into different equivalence classes or partitions. Remember to include both valid and invalid partitions.

**Boundary Value Analysis:** Identify the boundary values of the defined partitions and come up with test cases that include these boundary values. Ensure you consider "**edge cases**" - values just outside of valid ranges.

**Note:** Keep in mind that testing should cover not only expected or valid inputs but also unexpected or invalid ones. Consider all possible scenarios that might be encountered in a real-world situation.

**Questions to Product Owner:**

1) What is the lower limit when the income falls into the **"low"** category?

2) What is the upper limit when the income falls into the **"high"** category?

3) What numbers does the system accept only integers or also decimal numbers?

4) If the system accepts decimal numbers, how many digits should there be after the decimal point?

5) What character will the system accept for the decimal symbol, point (.) or comma (,)?

6) What is the step of the system, because if the system accepts only whole numbers, the step will be one, but if it accepts decimal numbers, the step will be relative to the number of symbols after the decimal point?

* I accept that the lower income limit for the **"low"** income category is 0.
* I accept that the upper income limit for the **"high"** income category is infinity.
* I accept that the system can accept both integer and decimal numbers. The number of characters after the decimal point is two characters.

**Equivalence Partitioning Test Cases including invalid cases:**

|  |  |  |
| --- | --- | --- |
| **Test Case ID** | **Input** | **Expected Output** |
| TC01 | 750 | "low" |
| TC02 | 2250 | "mid" |
| TC03 | 5575 | "high" |
| TC04 | -3679 | "error" |

**Boundary Value Analysis Test Cases including invalid cases:**

|  |  |  |
| --- | --- | --- |
| **Test Case ID** | **Input** | **Expected Output** |
| TC01 | -0,01 | "error" |
| TC02 | 0 | "low" |
| TC03 | 0,01 | "low" |
| TC04 | 999,99 | "low" |
| TC05 | 1000 | "mid" |
| TC06 | 1000,01 | "mid" |
| TC07 | 2999,99 | "mid" |
| TC08 | 3000 | "high" |
| TC09 | 3000,01 | "high" |

**Unexpected/Invalid inputs:**

|  |  |  |
| --- | --- | --- |
| **Test Case ID** | **Input** | **Expected Output** |
| TC01 | empty field | error, warning message |
| TC02 | string | error, warning message |
| TC03 | special characters | error, warning message |
| TC04 | mathematical operations(addition, subtraction, multiplication, and division) | error, warning message |
| TC05 | number that has more characters after the decimal point | error, warning message |

## 2. Pairwise Testing - eCommerce Checkout Function

Assume you have a checkout function of an eCommerce application for testing. The function contains the following fields with their input values:

**Drop-down menu that contains 5 different shipping methods (input values – 1, 2, 3, 4, 5);**

**Radio button for gift wrapping (input values – Yes or No);**

**Checkbox for agreeing to terms and conditions (input values - Checked or Unchecked);**

**Place Order button (input values - Does not accept any value, only finalizes the order).**

**Your task is:**

1. Calculate how many would be the positive test cases, if you have to cover every single possibility?

|  |
| --- |
| **Your Answer: 20** |

2. Using Pairwise testing, reduce the number of necessary test cases.

|  |
| --- |
| **Your Answer: 10** |

3. We have only considered positive test cases so far. What about negative ones? Write at least 3 negative test cases.

|  |
| --- |
| Attempting to place an order with empty field for shipping method. |
| Attempting to place an order with empty field for gift wrapping. |
| Attempting to place an order with empty field for agreeing to terms and conditions. |
| Attempting to place an order with empty field for everything. |