# 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.

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

|  |  |  |
| --- | --- | --- |
| **Test Case ID** | **Input** | **Expected Output** |
| **TC01** | **500** | **"low"** |
| **TC02** | **899** | **“low”** |
| **TC03** | **1000** | **“mid”** |
| **TC04** | **1500** | **“mid”** |
| **TC05** | **1998** | **“mid”** |
| **TC06** | **2000** | **“mid”** |
| **TC07** | **2035** | **“mid”** |
| **TC08** | **2890** | **“mid”** |
| **TC09** | **2999** | **“mid”** |
| **TC10** | **3000** | **“high”** |
| **TC11** | **3200** | **“high”** |
| **TC12** | **3500** | **“high”** |
| **TC13** | **4000** | **“high”** |
| **TC14** | **5670** | **“high”** |
| **TC15** | **-500** | **“error”** |
| **TC16** | **.** | **” Please use only numbers”** |
| **TC17** | **jkfdkdf** | **” Please use only numbers”** |
| **TC18** | **23.5** | **“Please enter whole numbers only”** |

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

|  |  |  |
| --- | --- | --- |
| **Test Case ID** | **Input** | **Expected Output** |
| **TC19** | **999** | **"low"** |
| **TC20** | **999.99** | **"low"** |
| **TC21** | **1000** | **“mid”** |
| **TC22** | **2999** | **“mid”** |
| **TC23** | **3000** | **“high”** |
| **TC24** | **0** | **“low”** |
| **TC25** | **Empty space** | **“Please, fill the space”** |
| **TC26** | **-1** | **“Error, not possible negative number”** |
| **TC27** | **-100** | **“Error, not possible negative number”** |
| **TC28** | **999lowincome** | **“Please, fill only number”** |
| **TC29** | **0.001** | **“Please, enter a valid value. The nearest possible is 0.** |

## 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:40** |

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Add a screenshot of the reduced test cases here**   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | **Shipping Methods** | **Gift Wrapping** | **Terms and Conditions** | **Place order Button** | | **1** | 3 | no | Unchecked | Only finalize the order | | **2** | 4 | no | Checked | Only finalize the order | | **3** | 4 | yes | Unchecked | Does not accept any value | | **4** | 5 | yes | Checked | Does not accept any value | | **5** | 5 | no | Unchecked | Only finalize the order | | **6** | 1 | no | Checked | Only finalize the order | | **7** | 1 | yes | Unchecked | Does not accept any value | | **8** | 2 | yes | Checked | Does not accept any value | | **9** | 2 | no | Unchecked | Only finalize the order | | **10** | 3 | no | Checked | Only finalize the order | | **11** | 3 | yes | Unchecked | Does not accept any value | | **12** | 4 | yes | Checked | Does not accept any value | | **13** | 4 | no | Unchecked | Only finalize the order | | **14** | 5 | no | Checked | Only finalize the order | | **15** | 5 | yes | Unchecked | Does not accept any value | |

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Shipping Methods** | **Gift Wrapping** | **Terms and Conditions** | **Place order Button** |
| **1** | 1 |  | Checked | Does not accept any value |
| **2** | 1 | yes |  | Only finalize the order |
| **3** | 1 | yes | Unchecked |  |
| **We have negative test when we provoke the system with leaving blank space on the mandatory fields.** | | | | |

|  |
| --- |
| **Example: Attempt to place an order with no shipping method selected.**  **Input: No shipping method selected**  **Expected Behavior: Error message prompt, not technically possible** |
| **Terms and Condition Not Checked**  **Input: Attempt to place an order without checking terms and conditions**  **Expected Behavior: The software should prevent placing and order without choosing terms and conditions** |
| **No selection of Gift Wrapping**  **Input: Leave the “Gift Wrapping” Button unmarked**  **Expected Behavior: The system should prompt the user to make selection of gift wrapping** |
| **Place an order without filling the mandatory fields**  **Input: Attempt to place an order without filling “Shipping Method”, “Gift Wrapping” and “Terms and Conditions”.**  **Expected Behavior: The system should display an error message for not filling all the mandatory fields.** |