 Progressive Education Society's

**Modern College of Engineering, Pune**

**MCA Department**

**A.Y.2023-24**

**(410907) Software Testing Laboratory**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Class: SY-MCA Shift / Div : S1 / B Roll Number : 52119

Name: Harsh Ghodke Assignment No: 3 Date of Implementation: 19 / 09 / 2024

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Q1 Write black box test cases using ECP & BVA to test the requirement –**

**Bank has different charges depending on the transaction done**

**5% of the amount for transaction less than or equal to 1000**

**6% of the amount for transaction more than 1000 and less than or equal to 2000 7% of the amount for transaction more than 2000**

**>>**

**Requirement:**

* 5% charge for transactions ≤ 1000
* 6% charge for transactions > 1000 and ≤ 2000
* 7% charge for transactions > 2000

•

**Equivalence Class Partitioning (ECP):**

**Equivalence Classes:**

* **Class 1 (≤ 1000)**: Valid partition where transaction amount is less than or equal to 1000.
* **Class 2 (> 1000 and ≤ 2000)**: Valid partition where transaction amount is between 1001 and 2000.
* **Class 3 (> 2000)**: Valid partition where transaction amount is greater than 2000

**2. Boundary Value Analysis (BVA):**

**Boundaries:**

* The boundary between Class 1 and Class 2: 1000 and 1001.
* The boundary between Class 2 and Class 3: 2000 and 2001

**Test Cases using ECP & BVA :**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case ID |  |  | Partition | Type of Input | Test Data | Expected Output |
| TC\_001 |  |  | Number between 1 to 1000 | Valid | 1, 500, 1000 | 5% Charge |
| TC\_002 |  |  | Number below 1 | Invalid | 0, -50 | Error Message |
| TC\_003 |  |  | Number between 1001 and 2000 | Valid | 1001, 1500, 2000 | 6% Charge |
| TC\_004 |  |  | Number greater than 2000 | Valid | 2001, 3000, 5000 | 7% Charge |
| TC\_005 |  |  | Boundary between 1000 and 1001  (BVA) | Valid | 1000, 1001 | 5% for 1000, 6% for 1001 |
| TC\_006 |  |  | Boundary between 2000 and 2001 (BVA) | Valid | 2000, 2001 | 6% for 2000, 7% for 2001 |

**Q 2. Write black box test cases using state based(state transition ) technique to test the requirement –**

**Withdrawal of money from ATM. ‘User A’ wants to withdraw 30,000 from ATM. Imagine he could take 10,000 per transaction and total balance available in the account is 25,000. In the first two attempts, he could withdraw money. Whereas in the third attempt, ATM shows a message as “Insufficient balance, contact Bank”. Same Action but due to change in the state, he couldn’t withdraw the money in the third transaction.**

**>>>**

**Scenario:**

* User A has ₹25,000 in their account.
* The user attempts to withdraw ₹30,000 in total, but is limited to ₹10,000 per transaction.
* The ATM allows two successful withdrawals, but on the third attempt, the balance is insufficient.

**States:**

1. S1: Initial state, balance = ₹25,000.
2. S2: After first withdrawal, balance = ₹15,000.
3. S3: After second withdrawal, balance = ₹5,000.
4. S4: Insufficient balance state after third attempt, message shown: “Insufficient balance, contact Bank”.

**Transitions:**

* T1: User successfully withdraws ₹10,000.
* T2: User successfully withdraws another ₹10,000.
* T3: User attempts to withdraw again but balance is insufficient.

**Test Cases using State Transition Technique:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case ID | Current State | Action (Input) | Next State | Expected Output |
| TC\_001 | S1 (₹25,000) | Withdraw ₹10,000 | S2 | Successful withdrawal, balance = ₹15,000 |
| TC\_002 | S2 (₹15,000) | Withdraw ₹10,000 | S3 | Successful withdrawal, balance = ₹5,000 |
| TC\_003 | S3 (₹5,000) | Withdraw ₹10,000 | S4 | Error: “Insufficient balance, contact Bank” |
| TC\_004 | S4 (₹5,000) | Check Balance | S4 | Balance = ₹5,000 |
| TC\_005 | S4 (₹5,000) | Attempt further withdrawal | S4 | Error: “Insufficient balance, contact Bank” |

**Q 3, Consider the following recursive code to find GCD of two numbers**

**findgcd (x , y)**

**{ if (x = y )**

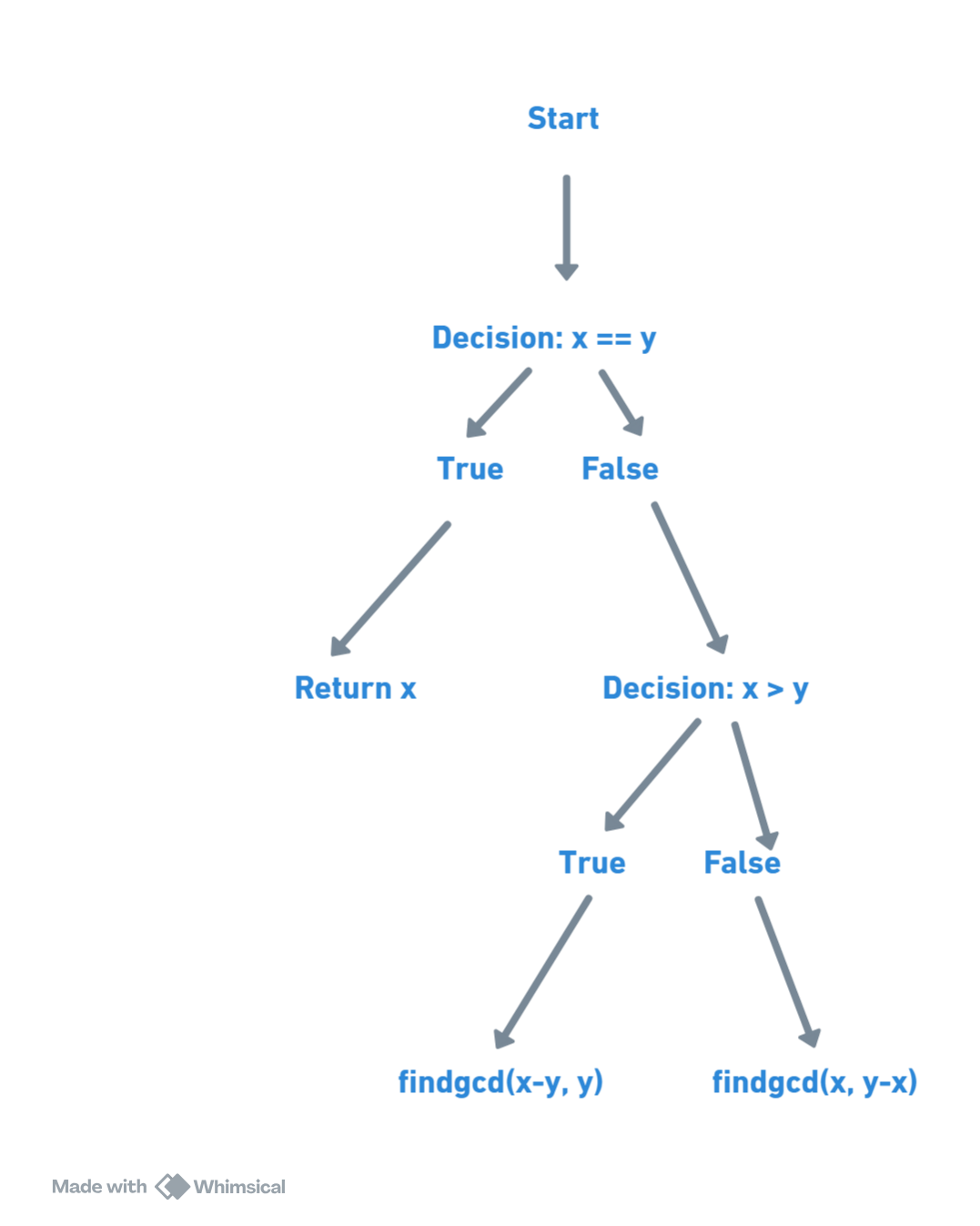
**return x; else if (x&gt;y) return findgcd(x-y,y); else return findgcd(x,y-x);**

**}**

**Draw flow graph, find cyclomatic complexity and derive test cases**

**>>>>**

**flow graph will look like this:**



**Cyclomatic Complexity**

The formula for cyclomatic complexity is:

V(G)=E−N+2

From the flow graph:

* Nodes (N): There are 4 nodes (Start, Decision 1, Decision 2, Return).
* Edges (E**)**: There are 5 edges (Start → Decision 1, Decision 1 → Return, Decision 1 → Decision 2, Decision 2 →

findgcd(x - y, y), Decision 2 → findgcd(x, y - x)).

V(G)=5−4+2=3V

**Thus, the cyclomatic complexity is 3.**

**Test Cases :**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Input (x, y)** | **Expected Output** | **Description** |
| TC\_001 | (15, 5) | 5 | x > y, takes path findgcd(x - y, y) multiple times, results in x == y. |
| TC\_002 | (7, 7) | 7 | x == y, base case, returns x. |
| TC\_003 | (14, 21) | 7 | x < y, takes path findgcd(x, y - x) multiple times, results in x == y. |

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***