Question No: 01

[4\*2.5 marks]

You are testing a photo-enforcement system for traffic control in an intersection. Consider the following scenarios

(Scenario i) A photo will be taken if the light is red (RED) and the front wheels of the car are over the line marking the beginning of the intersection (WHEELS).

a) Which sets of values provides the minimum tests to achieve 100% decision/condition coverage?

(Scenario ii) A photo should be taken if the signal light is red (RED) or the car is speeding (SPEED) and if the front wheels of the car are over the line marking the beginning of the intersection (WHEELS).

b) Which sets of values provide the minimum tests to achieve 100% modified condition/decision coverage?

c) Which sets of values provide the minimum tests to achieve 100% multiple condition coverage?

d) Which sets of values provide the minimum tests to achieve 100% path coverage?

Ans:

a)

Variables – RED and WHEEL (Condition)

Action – PHOTO TAKEN

2^2 = 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Test case 1 | Test case 2 | Test case 3 | Test case 4 |
| RED | F | T | F | T |
| WHEEL | F | F | T | T |
| PHOTO TAKEN | F | F | F | T |
|  |  |  |  |  |

Condition should be evaluated to condition 1 and 4

b)

(Scenario ii) A photo should be taken if the signal light is red (RED) or the car is speeding (SPEED) and if the front wheels of the car are over the line marking the beginning of the intersection (WHEELS).

b) Which sets of values provide the minimum tests to achieve 100% modified condition/decision coverage?

2^n, n is action

2^3 = 8 (2 to power of 3)

Variables – RED, SPEED and WHEEL (Condition)

Action – PHOTO TAKEN

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Test case 1 | Test case 2 | Test case 3 | Test case 4 | Test case 5 | Test case 6 | Test case 7 | Test case 8 |
| RED | F | F | F | F | T | T | T | T |
| SPEED | F | F | T | T | F | F | T | T |
| WHEEL | F | T | F | T | F | T | F | T |
| PHOTO TAKEN | F | F | F | T | F | T | F | T |

Test 2 and 6 is validate RED case

Test 2 and 4 is validate SPEED case

Test 4 and 7 is validate WHEEL case

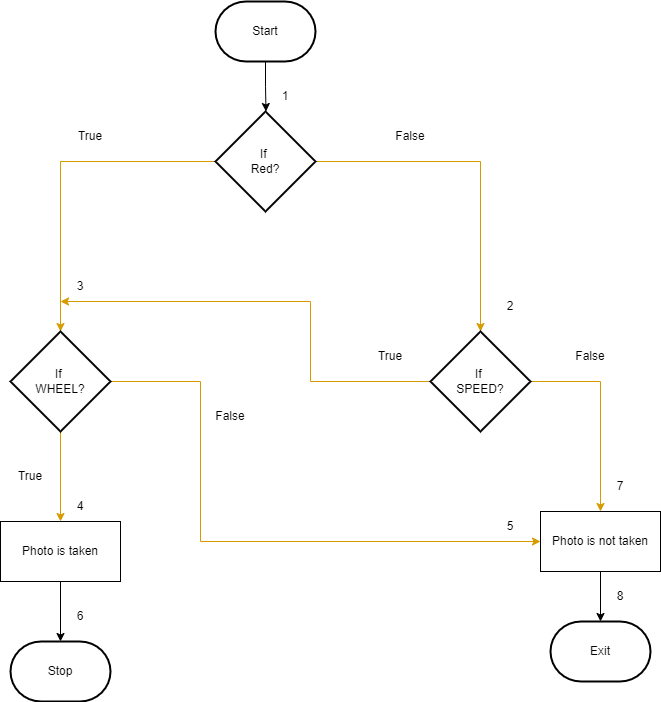
1,2,4 & 6 minimum test is required

Note: number of test case is n+1, where n is number of conditions

c)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Test case 1 | Test case 2 | Test case 3 | Test case 4 | Test case 5 | Test case 6 | Test case 7 | Test case 8 |
| RED | F | F | F | F | T | T | T | T |
| SPEED | F | F | T | T | F | F | T | T |
| WHEEL | F | T | F | T | F | T | F | T |
| PHOTO TAKEN | F | F | F | T | F | T | F | T |

d) Which sets of values provide the minimum tests to achieve 100% path coverage?



Path coverage – 1, 2, 3 and 8

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Test case 1 | Test case 2 | Test case 3 | Test case 4 | Test case 5 | Test case 6 | Test case 7 | Test case 8 |
| RED | F | F | F | F | T | T | T | T |
| SPEED | F | F | T | T | F | F | T | T |
| WHEEL | F | T | F | T | F | T | F | T |
| PHOTO TAKEN | F | F | F | T | F | T | F | T |

Question No. 02

An IOT enabled bulb changes states between red, green and blue on receiving the following input signals via Wi-Fi

[6 +4 marks]

Enable = R255 ------ Red state

Enable = G255 ------ Green state

Enable = B255 ------ Blue state

(i) In case of other input being sent at the Enable - the bulb takes a 'Dim state'.

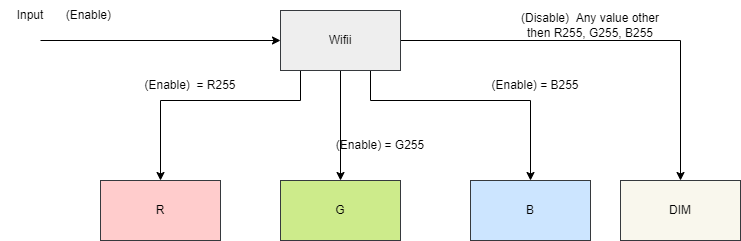
a) Draw the state transition diagram for the bulb.

b) Write test cases to test all the states.

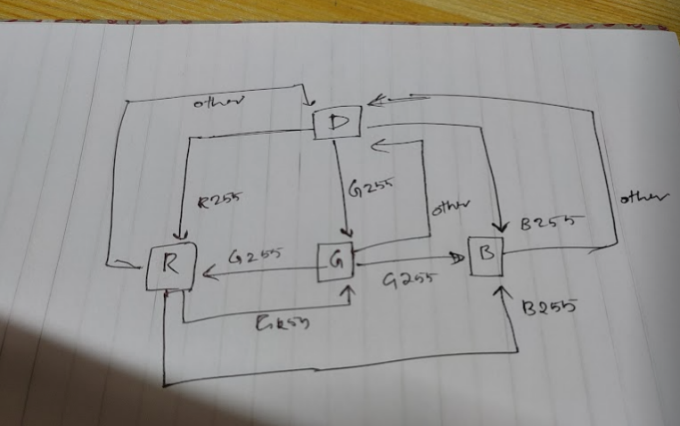
(ii) The bulb needs to interact with a Wi-Fi enabled device in order to receive the 'enable' input. Write system-level test cases to test this system.

Ans:

a)



Other reference



b)

Ans:

C1: R

C2: G

C3: B

A1: Bulb is ON state

A2: Bulb is Dim state

2^3(conditions) = 2\*3= 8 (Rules)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Conditions | Rule1 | Rule2 | Rule3 | Rule4 | Rule5 | Rule6 | Rule7 | Rule8 |
| R | F | F | F | F | T | T | T | T |
| G | F | F | T | T | F | F | T | T |
| B | F | T | F | T | F | T | F | T |
| Action |  |  |  |  |  |  |  |  |
| A1 | F | T | T | T | T | T | T | T |
|  |  |  |  |  |  |  |  |  |

(ii) System-Level Test Cases to Test the Wi-Fi Interaction:

**Chatgpt answer**

1. Test Case 1: Verify that the bulb successfully connects to the Wi-Fi network.  
2. Test Case 2: Send valid "Enable" input signals (R255, G255, B255) through Wi-Fi and verify that the bulb transitions to the corresponding states (Red, Green, Blue) as expected.  
3. Test Case 3: Send an invalid "Enable" input signal through Wi-Fi and verify that the bulb remains in the "Dim" state.  
4. Test Case 4: Test the stability and reliability of the Wi-Fi connection by continuously sending valid "Enable" input signals and ensuring consistent state transitions.  
5. Test Case 5: Test the response time of the bulb to the Wi-Fi signals by measuring the time taken for the bulb to transition to the corresponding state after receiving an "Enable" input.  
  
These test cases cover different scenarios and interactions between the bulb and the Wi-Fi-enabled device, ensuring the functionality and reliability of the system.

Question No: 03

Consider the following algorithm that calculates the sum of all even natural numbers between 1 to N. Answer the following questions:

int evenNaturalSum(int N)

{

int sum = 0;

int count = 1;

while(count<=N)

{

if (count%2==0)

{

sum = sum + count;

}

count++;

}

return sum;

}

a) Assume that the range of signed int is 2^8=256. Then N can take values only in range 0 to 255 [including end points]. Calculate the number of test cases - Normal BVA, Robust BVA, Worst case BVA and Robust worst case BVA.

b) Write all the test cases for Robust BVA for the above algorithm.

c) Identify all DU pairs for the variables sum, count and N. For all the use nodes, specify whether it is a C-Use or P-Use and why?

Ans:

Sum of even number from 1 – 255 =16256+255 =**16511**

BVA=4n + 1 = 13

Robust BVA = 6n +1 = 19

Worst case BVA= 5^n = 126

Robust Worst case BVA= 7^n = 343

* n = number of inputs
* Here n is 3 (conditions)

Question 4

Select a structure-based technique according to a given project situation and justify your answer.

You are the Senior Test Analyst working on the testing of Air traffic control software that helps to prevent collisions, organize and expedite the flow of air traffic, and provide information and other support for pilots. A failure analysis has shown that if the software system fails then it may risk the life of thousands of travelers. The government has requested that the level of testing for this software exceeds that normally required by the relevant regulatory standards.

Which is the level of test coverage you would expect to be achieved in of the control software? Also, justify your answer. [5 marks]

A. Multiple Condition coverage

B. Branch coverage + Modified Condition/Decision coverage

C. Branch coverage + Statement coverage

D. Modified Condition/Decision coverage

Ans:

Given the critical nature of the Air Traffic Control software and the potential risk to the lives of thousands of travelers, it is crucial to ensure thorough testing to achieve a high level of test coverage. Considering the severity and impact of potential failures, the appropriate choice would be:

B. Branch coverage + Modified Condition/Decision coverage

Justification:

1. Branch Coverage: Branch coverage ensures that all decision outcomes (branches) in the software are tested at least once. In the context of Air Traffic Control software, it is vital to verify that all possible decision paths and control flows are exercised to identify potential flaws or errors in the logic. By achieving branch coverage, we can have more confidence that critical decision points are tested adequately.

2. Modified Condition/Decision Coverage (MC/DC): MC/DC is a technique that ensures each condition in a decision statement independently affects the decision outcome. It focuses on testing different combinations of conditions, ensuring that each condition has been tested both when it evaluates to true and when it evaluates to false. MC/DC helps uncover complex interdependencies and potential issues related to conditions and decision outcomes. Given the criticality of the software and the need to prevent collisions and ensure the safe flow of air traffic, MC/DC would be an appropriate choice to ensure thorough testing of condition combinations.

The combination of Branch coverage and Modified Condition/Decision coverage provides a strong assurance that critical decision points are covered and that conditions within those decisions are adequately tested. This approach helps identify potential flaws or errors in the software, reducing the risk of failures that could endanger the lives of travelers.

Question 5

Class Integer Stack provides implementation of Stack data structure, with the implementation of two methods: push() and pop(). Moreover, classes OddIntegerStack and EvenIntegerStack extend the class IntegerStack and overrides the implementation of the push() method. Class Odd Integer Stack is shown below:

public class OddIntegerStack extends IntegerStack{

public void push (int element)

{

if (element % 2 == 0)

{

System.out.println("Cannot push even element onto an odd stack");

return;

}

If (top != maxSize-1){

arrayStack [++top] =element;

}

else{ System.out.println("Overflow error");}

}

}

[2+2+3+3 marks]

a) Write test cases to test the push()method of the Odd IntegerStack class. Assume maxSize =5

b) Write test cases to test the pop()method of the OddIntegerStack class. Assume maxSize = 5.

c) An object reference of class IntegerStack can be used to store an object of class

Even Integer Stack or OddIntegerStack. Write test cases to test this polymorphic behaviour.

d) Discuss, in brief, how you would carry out integration testing of such a system.