

Birla Institute of Technology & Science, Pilani
Second Semester 2022-2023
Comprehensive Examination

Course Title : SOFTWARE TESTING METHODOLOGIES
 Weightage : 45 Marks + 10
 Duration : 3 Hours

Q1 Given a graph with nodes $N = \{n_0, n_1, n_2, n_3, n_4, n_5, n_6, n_7, n_8, n_9\}$; initial nodes $N_0 = \{n_0, n_1, n_2\}$, accepting nodes $N_f = \{n_7, n_8, n_9\}$ and edges $E = \{(n_0, n_3), (n_0, n_4), (n_1, n_4), (n_2, n_5), (n_2, n_6), (n_3, n_7), (n_4, n_7), (n_4, n_8), (n_5, n_1), (n_5, n_9), (n_6, n_9), (n_8, n_5)\}$. List test paths that achieve Node Coverage but not Edge Coverage on the graph. [2]

Q2 For the graph below, answer the following questions:

$N = \{1, 2, 3, 4, 5, 6\}$
 $N_0 = \{1\}$
 $N_f = \{6\}$
 $E = \{(1, 2), (2, 3), (2, 6), (3, 4), (3, 5), (4, 5), (5, 2)\}$
 $def(1) = def(5) = use(5) = use(6) = \{x\}$
 // Assume the use of x in 5 precedes the def
Test Paths:
 $t_1 = [1, 2, 6]$
 $t_2 = [1, 2, 3, 4, 5, 2, 3, 5, 2, 6]$
 $t_3 = [1, 2, 3, 5, 2, 3, 4, 5, 2, 6]$

- List a minimal test set that satisfies all defs coverage with respect to x. [2]
 - List a minimal test set that satisfies all uses coverage with respect to x. [2]
 - List a minimal test set that satisfies all du-paths coverage with respect to x. [2]
- (d) Obtain all du-path w.r.t variable x [5]
- Q2.** Consider a finite state machine (FSM) as shown Figure 1.

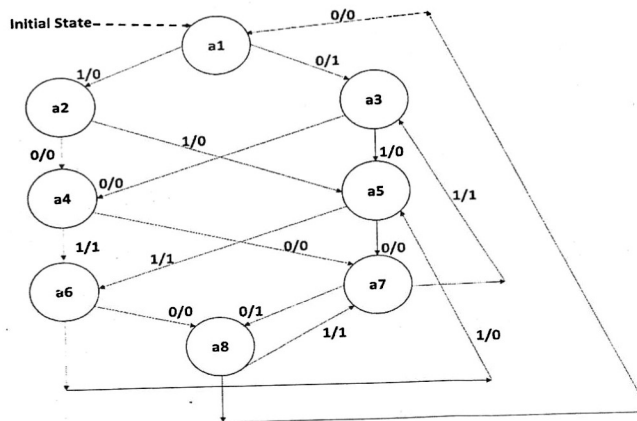
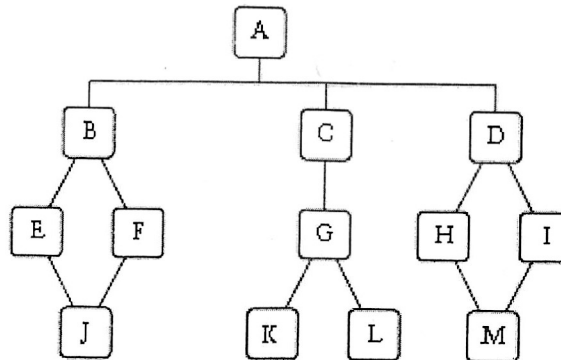


Figure 1: Finite State Machine

Explain how you will generate the test cases for the FSM as shown in Figure 1 using W Method (Assume number of states in both the correct implementation and the implementation under test IUT are same). Please follow the steps sequentially and show the output of each step clearly. [8] +2

Q3. Using the structure chart shown below,



Specify the no. of sessions and no of stubs/drivers using top-down integration. [2]

+ 2
& bottom up integration

Q4. Given below a program fragment with test inputs. Draw the call graph. Find a minimal test set that achieves Node Coverage. Find a minimal test set that achieves Edge Coverage.

```

public static void f1 (int x, int y)
{
    if (x < y) { f2 (y); } else { f3 (
}
public static void f2 (int a)
{
    if (a % 2 == 0) { f3 (2*a); };
}
public static void f3 (int b)
{
    if (b > 0) { f4(); } else { f5();
}
  
```

Use the following test inputs:

- t1 = f1 (0, 0)
- t2 = f1 (1, 1)
- t3 = f1 (0, 1)
- t4 = f1 (3, 2)
- t5 = f1 (3, 4)

a. Find a minimal test set that achieves Node Coverage. [2].

b. Find a minimal test set that achieves Edge Coverage. [2].

Q5. The following code snippet shows creating an object of class Date. Date class internally uses class Day, class Month and Class Year to deliver the functionalities as discussed in lecture.

```

class testIt
main()
testdate = instantiate Date(testMonth, testDay, testYear)
testdate.increment()
testdate.printDate()
End 'testIt
  
```

a. Extend the Date class to support zodiac signs for every date. [2]

b. Draw the collaboration diagram for the entire class hierarchy. [2]

c. Draw the sequence diagram for setting a date along with a specific zodiac sign. [2]

- d. Depict the MM path for setting a specific date and zodiac sign. (consider all corner cases) [2].

Q6. The calculator application converts numerical inputs: log value, factorial, prime or not, Fibonacci value till that input. The user can revise inputs and perform repeated calculation

Tasks:

- List the system functions and categorize them. [2].
- List the HUC, EUC, EEUC and RUCs for the above system. (UI design included) [2].
- Draw the incidence matrix for the above scenario. [1].

Q7. Consider the following java code snippet to compute the factorial of a number.

```
1 static int factorial(int n){
2     if (n == 0)
3         return 1;
4     else
5         return(n * factorial(n-1));
}
```

- Write the JUnit tests for it "Assuming it should return 0 if there is an error" [2]
- Did the code pass all your tests or not? If yes, why? If no, explain possible fail cases? [2]
- Can you completely test this factorial function? Justify your answer. [2]

You were hired by a local software development company as security consultant. As a first task, you should analyze the following programme:

```
1
2 public class UserCount {
3
4     public static boolean duplicateUser(String username) throws Exception
5
6         // ...
7         String sqlStatement = "SELECT COUNT(*) FROM user_table where use
            username;
8         // ...
9         int count = jdbcConnection.execute(sqlStatement);
10        if(count <= 1) {
11            return true;
12        }
13        return true;
14    }
15 }
16
17 public class Main {
18
19     public void doSomething() {
20         // ...
21         // POST("ATTRIBUTE_NAME") receives unfiltered HTTP post requ
            client
22         String username = POST("username")
23         UserCount.duplicateUser(username);
24         // ...
25     }
26 }
```

Tasks :-

- d. Name the vulnerability and briefly describe how an attacker can potentially exploit the code above [2].
- e. Briefly explain how the vulnerability can be fixed. [2].
