Birla Institute of Technology & Science, Pilani
Mid term Examination

Course Title : SOFTWARE TESTING METHODOLOGIES
Weightage : 30% (As per Course Handout)
Duration : 1.5 Hours

Q1 At least how many test cases are required to achieve multiple condition coverage of the following code segment: **[1]**

If ((a>5) and (b<100) and (c>50)) x=x+1;

a) 2

b) 4

c) 6

d) 8

Q2. For a function of 4 variables, boundary value analysis generates: **[1]**

a) 9 test cases

b) 17 test cases

c) 33 test cases

d.) 25 test cases

Q3. Consider the following program:

```
while (first <= last)
{
  if (array [middle] < search)
    first = middle +1;
  else if (array [middle] == search)
    found = True;
  else last = middle – 1;
  middle = (first + last)/2;
}
if (first < last) not Present = True;
```

The cyclomatic complexity of the program segment is __________ **[1]**

a. 3
b. 4
c. 5
d. 6

Q4. Which of the following statements about the relationship of statement coverage and decision coverage is correct? **[1]**

a. 100% statement coverage means 100% decision coverage.

b. 100% decision coverage means 100% statement coverage.

c. 90% statement coverage means 90% decision coverage.

d. 90% decision coverage means 90% statement coverage.

Q5. Find if the following statements in the context of software testing are TRUE or FALSE. **[1]**

(S1) Statement coverage cannot guarantee execution of loops in a program under test.

(S2) Use of independent path testing criterion guarantees execution of each loop in a program under test more than once.

a. True, False

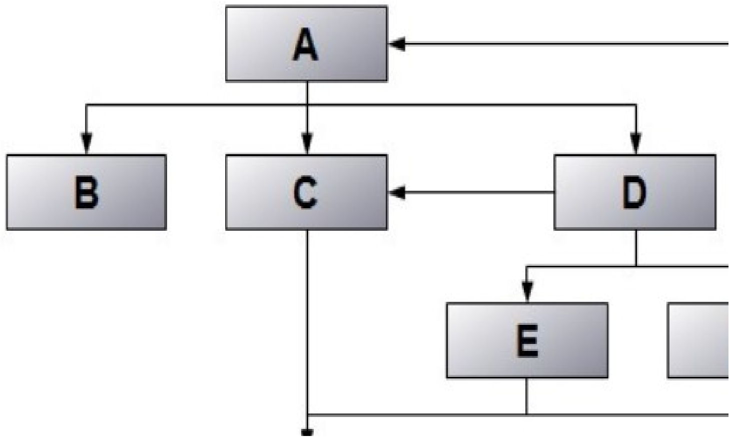
b. False, True

c. True, True

d. False, False

Q6. For the graph below (N is the set of nodes, N0 is initial node, Nf is the accepting node, E represents the edges):
• N = {1, 2, 3} • N0 = {1} • Nf = {3} • E = {(1, 2), (1, 3), (2, 1), (2, 3), (3, 1)}
Also consider the following (candidate) paths: • p1 = [1, 2, 3, 1] • p2 = [1, 3, 1, 2, 3] • p3 = [1, 2, 3, 1, 2, 1, 3] • p4 = [2, 3, 1, 3] • p5 = [1, 2, 3, 2, 3]
Draw the graph. Which of the listed paths are test paths? **[3]**

Q7. For the following CFG, obtain the following: Write down which paths the required tests should cover to achieve 100 % Statement Coverage. Write down, which paths the required tests should cover to achieve 100 % Decision Coverage. Calculate the cyclomatic complexity



[3]

Q8. Comment with Yes/ No for the following statements. Provide a justification to each one of them with a sample code and justification. **[4]**

a. Does 100% all-use coverage gives 100% multiple-condition coverage?
b. Does 100% p-use coverage gives 100% decision coverage?

Q9. Suppose that coverage criterion C1 subsumes coverage criterion C2. Further suppose that test set T1 satisfies C1 on program P and test set T2 satisfies C2, also on P. Does T1 necessarily satisfy C2? Explain. Does T2 necessarily satisfy C1? Explain. **[4]**

Q10. Consider a path *'s'* through some program *'P'*. Variable *'v'* is defined along this path *'s'* at some node N_d and used subsequently at some node N_u in an assignment i.e. there is a c-use of variable *'v'* along the path *'s'*. [Note: In the following two questions Q11 (a) and Q11 (b), simply writing True/False or Yes/No will fetch zero mark. You have to give a proper example to justify your choice.]

i. Suppose now that path *'s'* is infeasible. Does this imply that c-use of *'v'* at node *Nu* is also infeasible. Explain your answer with the help of a suitable example. **[2]**

ii. Suppose now that there is a p-use of variable *'v'* at some node N_p along the same path 's'. Given that path *'s'* is infeasible, Is this p-use of the variable *'v'* also infeasible. Explain your answer with the help of a suitable example. **[2]**

iii. Consider a program 'P' and its first-order mutant 'M' as shown in the following table. Actually as per the requirements, the program 'P' has an error at line 3 and the generated first order mutant 'M' is the actual correct version. Construct a test case T_1 that causes P to fail and establishes the relation $P(T_1) \neq M(T_1)$. Can you construct a test case T_2 on which P is successful but $P(T_2) \neq M(T_2)$. **[5]**

Program P	Mutant M
1. input x, y	1. input x, y
2. if x < y Then	2. if x < y Then
3. z = x * (y + x);	3. z = x * (y + 1);
4. else	4. else
5. z = x * (y -1);	5. z = x * (y -1);

Q11. Identify the test suites between T1, T2, T3 and T4 that ensures statement coverage for the following program**[2]**

```
Begin
If (x==y) {P1; exit;}
else if {u==v} {P2;}
else {P3; exit;}
P4;
End
```

T1: x=y and x!= v
T2: x!=y and u=v
T3: x,y,u and v are all distinct
T4: x,y,u and v are all equal