**Fault & Failure Model**

Three conditions necessary for a failure to be observed

1. Reachability : The location or locations in the program that contain the fault must be reached
2. Infection : The state of the program must be incorrect
3. Propagation : The infected state must propagate to cause some output of the program to be incorrect

|  |
| --- |
| public int findLast (int[] x, int y)  {  //EFFECTS: If x is empty, return -1  //else returns the index of the last element in x that equals to y  //if no such element exists, return -1  if (x.length ==0)  return -1;  for (int i=x.length-1;i>0;i--){  If (x[i]==y)  return i;  }  return -1;  } |

1. Identify the fault.
2. Identify a test case,
   1. That doesn’t reach the faulty statement.
   2. That executes the faulty statement, but not failed.
   3. That reveals failure

Refer the following code and answer the given questions.

|  |
| --- |
| /\*This method returns the index of **last occurrence** of the maximum value in the given array. It expects an array having atleast 5 elements. If that condition fails, it returns -1.  Eg: if numbers=[1,2,5,3,5,2], it should return 4.  \*/  public static int getMaxValueIndex(int[] numbers){ //NODE 1  if (numbers.length<5) //NODE 1  return -1; //NODE 2  int maxValue = numbers[numbers.length-1]; //NODE 3  int indx = numbers.length-1; //NODE 3  for(int i= numbers.length-2;i>=0;i--){ //NODE 3  if(numbers[i] >= maxValue){ //NODE 4  maxValue = numbers[i]; //NODE 5  indx = i; //NODE 5  }  }  return indx; //NODE 6  } //NODE 6 |

1. **“Faults do not result in Failures always.”** Justify above statement explain the conditions to satisfy for a Failure to occur. (4 marks)
2. Identify the **fault** in above given code fragment. (2 marks)
3. Identify a Test Case, (6 marks)
   1. That doesn’t execute the faulty statement.
   2. That executes the fault but doesn’t result in Infection.
   3. That result in Infection, but not propagated.
   4. That reveals a Failure.