TECH MODULE OBJECTIVE : MINI-PROJECT(S)

<table>
<tr><th>S No.</th><th>MINI-PROJECT DESCRIPTION</th></tr>
<tr><td>1</td><td>This is a simplemathcomputer where individually 3 modules has been done. You as a unit tester and bug fixer would complete below 3 testcases individually and then create a testsuite add all three modules under one testsuite and test the same.</td></tr>
<tr><td>1.1</td><td>

Test the following code using JUNIT.

The following program is written to test whether the given number is Cube or Not.

```
public boolean isCube(int N){
     boolean status=false;
           int j,k,c1=0;
            j=N/2;
            int c=0;

     if(N==1)   {
          status=true;   return status;
     }

     for(k=2;k<=j;k++)    {
               if(N%k==0)  {
                              if((k*k*k)=N){
                              c1=1;
                              }
            }
}
    if(c1==1)    {
          status=true;
    }
    else    {
           status=false;
   }
    return status;
}
Correct the errors in the code and test the behavior with
JUNIT
```

</td></tr>
<tr><td>1.2</td><td>

The following program is written to find Nth Fibonacci number.

Note: Fibbonacci sequence: 0, 1, 1, 2, 3, 5, 8, 13, ….

```
public  int nthFibonacci(int num)    {
        if(num==1){
```

</td></tr>
</table>

```
            return 0;
        }
        if(num==2){
            return 1;
        }
        return nthFibonacci(num-1)+nthFibonacci(num+1);
    }
```

Test the function using JUNIT & check whether it is finding the Nth fibbonacci number.

Try to fix the logical errors in these functions.

1.3

```
public class Simputer {

//function1
// do not change the data types
byte add( byte b1 , byte b2){
return (byte)  (b1+b2);
}
//function2
// do not change the data types
short add( short b1 , short b2){
return (short) (b1+b2);
}
//function3
// do not change the data types
  int[] print_binary(byte number){
     int[] result;
     int size=10;

     if ( number< 16)
          result= new int[4];
     else if (number< 32 )
          result= new int[5];
     else if (number< 64 )
          result= new int[6];
     else
          result= new int[7];

     int i=0;
     for( i=0; number>=1; i++)     {
          result[i]=number%2;
                i++;
          number=(byte)(number/(byte)2);
```

```
        }
        result[i]=number;
        return result;
}

}
```

Test the above functions of simputer class using JUNIT.

print_binary function is used to print the binary equivalent of the given number.

Note: Try to fix the logical errors in these functions.

You need to use assertEquals, assertArrayEquals methods as needed.

```
@Test
public void test_print_binary() {
        int res1[] = {0,1,0,1};
        assertArrayEquals(res1,
Simputer.print_binary((byte)10));
        res1 = new int[] {1,1,0,1};
        assertArrayEquals(res1,
Simputer.print_binary((byte)11));
        }
```

Test all three functions using ParameterisedTest mechanism.