Smt. Chandaben Mohanbhai Patel Institute of Computer Applications
Master of Computer Applications (MCA)
Semester - II
CA870 Software Quality Assurance

Assignment-3 Unit Testing with Junit

1. Write a Program to find given number is prime number or not.

Check List:

- Any Element of an Input set is Negative
- Any Element of an Input set is Alphabetic
- Any Element of an Input set is Special Symbol
- Any Element of an Input set is Zero
- There are More than one input

List	Checklist	Input	Expected Output
1.	Element Of an Input is Negative	-5	Error Message: Negative Not Allowed
2.	Element Of an Input is Alphabetic	A	Error Message: Alphabets Not Allowed
3.	Element Of an Input is Special Symbol	&	Error Message : Special Symbol Not Allowed
4.	Element Of an Input is Zero	0	Error Message : zero Not Allowed
5.	There are More than one input	(25,5,10)	Error Message : More than one input Not Allowed

Junit test code:-

```
package Admin;


public class primeornot {
    public static boolean isPrime(int number) {
```

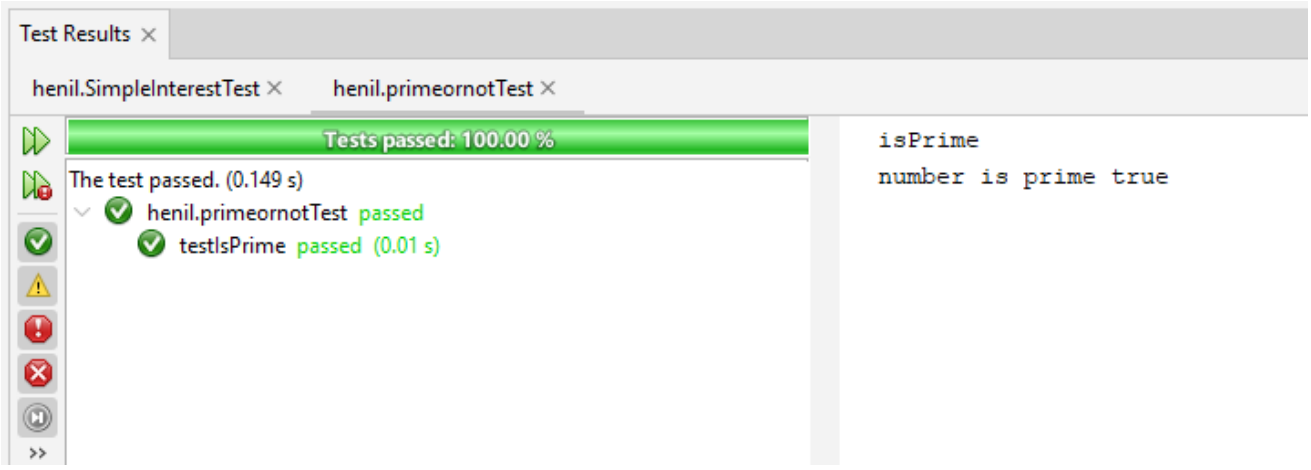
```
        if (number <= 1) {
            return false;
        }

        for (int i = 2; i <= number / 2; i++) {
            if (number % i == 0) {
                return false;
            }
        }

        return true;
    }
}
```

```
package Admin;

import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
import static org.junit.Assert.*;


public class primeornotTest {

    public primeornotTest() {
```

```
}

@BeforeClass
public static void setUpClass() {
}

@AfterClass
public static void tearDownClass() {
}

@Before
public void setUp() {
}

@After
public void tearDown() {
}

/**
 * Test of isPrime method, of class primeornot.
 */
@Test
public void testIsPrime() {
    System.out.println("isPrime");
    int number = 5;
    boolean expResult = true;
    boolean result = primeornot.isPrime(number);
```

```
        assertEquals(expResult, result);
        if(expResult==true){
                System.out.println("number is prime "+expResult);

        }else{
                System.out.println("number is not prime "+expResult);

        }
        // TODO review the generated test code and remove the default call to fail.
        //fail("The test case is a prototype.");
    }

}
```

Output:-



2. Write a program to calculate simple interest.

Check List:

- Any Element of an Input set is Negative
- Any Element of an Input set is Alphabetic

- Any Element of an Input set is Special Symbol
- Any Element of an Input set is Zero
- There are Less than three input
- There are More than three input

List	Checklist	Input (P,R,N)	Expected Output
1.	Any Element Of an Input set is Negative	(-1500,2,5), (1500,-1,1), (1600,1,-4)	Error Message: Negative Not Allowed
2.	Any Element Of an Input set is Alphabetic	(A,2,5),(1500,B,1), (1600,1,A)	Error Message: Alphabets Not Allowed
3.	Any Element Of an Input set is Special Symbol	(!,5,5), (1500,#,1), (1600,15,&)	Error Message : Special Symbol Not Allowed
4.	Any Element Of an Input set is Zero	(0,25,5), (1500,0,1), (1600,15,0)	Error Message : zero Not Allowed
5.	There are Less than three input	(1500,25)	Error Message : less than three Not Allowed
6.	There are More than three input	(1500,25,5,10)	Error Message : More than three input Not Allowed
7.	First Input Must Principle Amount, Second Input Must Rate Of Interest, Third Input Must Number Of Years	(2,25000,5)	Error Message : Invalid input
8.	Correct input of P,R,N	(1500,10,5)	750

Junit test code:-

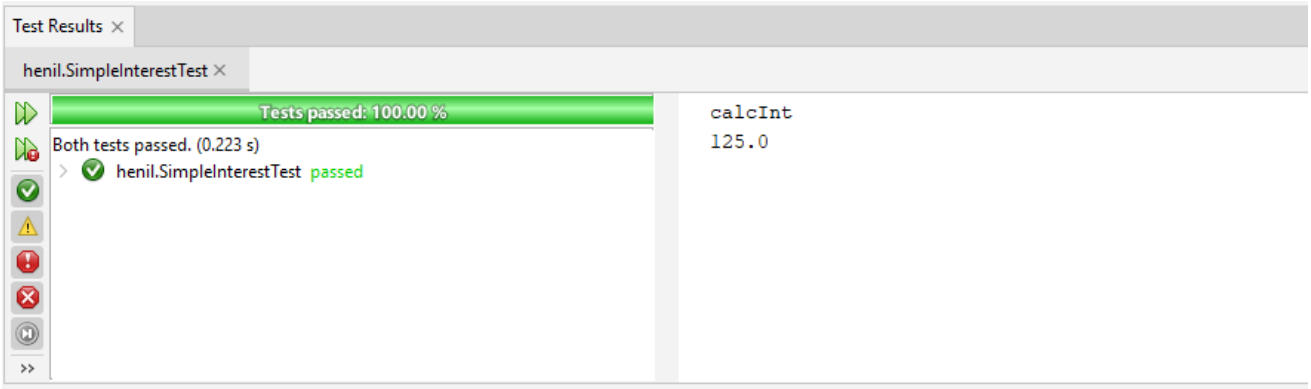
```
package Admin;

public class SimpleInterest {

    public float calcInt(int p,int r,int n){

        if(p<=0||r<=0||n<=0){
```

```
            System.out.println("Zero or Negative number is not allowed");
        }

        return ((p*r*n));
    }
}

package Admin;

import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
import static org.junit.Assert.*;


public class SimpleInterestTest {

    public SimpleInterestTest() {
    }

    @BeforeClass
    public static void setUpClass() {
    }

    @AfterClass
```

```
public static void tearDownClass() {
}

@Before
public void setUp() {
}

@After
public void tearDown() {
}

@Test
public void testSomeMethod() {
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}


@Test
public void testCalcInt() {
    System.out.println("calcInt");
    int p = 5;
    int r = 5;
    int n = 5;
    SimpleInterest instance = new SimpleInterest();
    float expResult = 125.0F;
    float result = instance.calcInt(p, r, n);
```

```
        assertEquals(expResult, result, 0);
        System.out.println(expResult);
        // TODO review the generated test code and remove the default call to fail.
        //fail("The test case is a prototype.");
    }

}
```

Outout:-



3. Write a program to generate Fibonacci series up to requested number.

Check List:

- Any Element of an Input set is Negative
- Any Element of an Input set is Alphabetic
- Any Element of an Input set is Special Symbol
- Any Element of an Input set is Zero
- There are More than one input

List	Checklist	Input	Expected Output
1.	Element Of an Input is Negative	-5	Error Message: Negative Not Allowed
2.	Element Of an Input is Alphabetic	A	Error Message: Alphabets Not Allowed

3.	Element Of an Input is Special Symbol	&	Error Message : Special Symbol Not Allowed
4.	Element Of an Input is Zero	0	Error Message : zero Not Allowed
6.	There are More than one input	(15,25,5,10)	Error Message : More than one input Not Allowed

Junit test code:-

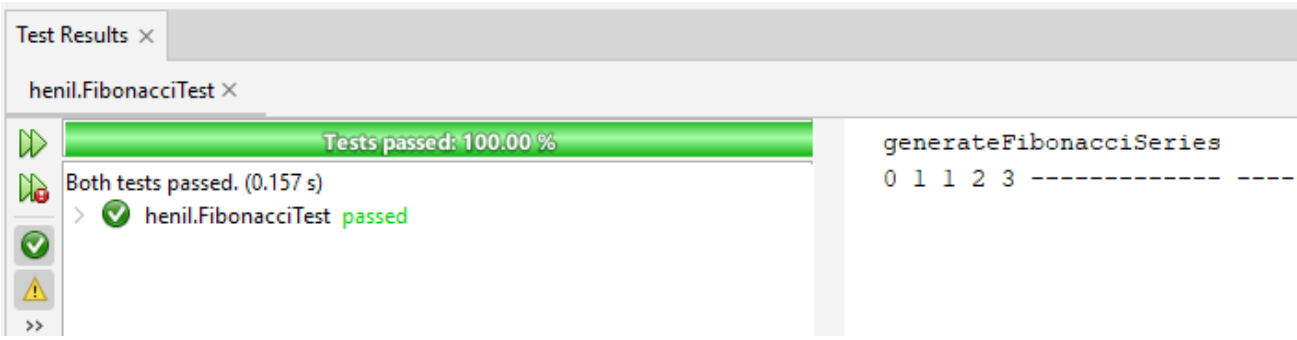
```
public class Fibonacci {
    public static int[] generateFibonacciSeries(int n) {
      int[] series = new int[n];
      series[0] = 0;
      series[1] = 1;

      for (int i = 2; i < n; i++) {
         series[i] = series[i - 1] + series[i - 2];
         if (series[i] > n) {
            break;
         }
      }

      return series;
   }
}

package Admin;
```

```
import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
import static org.junit.Assert.*;



public class FibonacciTest {

    public FibonacciTest() {
    }

    @BeforeClass
    public static void setUpClass() {
    }

    @AfterClass
    public static void tearDownClass() {
    }

    @Before
    public void setUp() {
    }

    @After
    public void tearDown() {
```

```
    }

    @Test
    public void testSomeMethod() {

        // TODO review the generated test code and remove the default call to fail.
        //fail("The test case is a prototype.");
    }



    @Test
    public void testGenerateFibonacciSeries() {
        System.out.println("generateFibonacciSeries");
        int n = 5;
        int[] expResult = {0,1,1,2,3};
        int[] result = Fibonacci.generateFibonacciSeries(n);
        assertArrayEquals(expResult, result);
        for (int term : expResult) {
            System.out.print(term + " ");
        }

        // TODO review the generated test code and remove the default call to fail.
       // fail("The test case is a prototype.");
    }

}
```

Output:-

Smt. Chandaben Mohanbhai Patel Institute of Computer Applications
Master of Computer Applications (MCA)
Semester - II
CA870 Software Quality Assurance



4. Write a program to find the factorial of a given number.

Check List:

- Any Element of an Input set is Negative
- Any Element of an Input set is Alphabetic
- Any Element of an Input set is Special Symbol
- Any Element of an Input set is Zero
- There are More than one input

List	Checklist	Input	Expected Output
1.	Element Of an Input is Negative	-5	Error Message: Negative Not Allowed
2.	Element Of an Input is Alphabetic	A	Error Message: Alphabets Not Allowed
3.	Element Of an Input is Special Symbol	&	Error Message : Special Symbol Not Allowed
4.	Element Of an Input is Zero	0	Error Message : zero Not Allowed
6.	There are More than one input	(15,25,5,10)	Error Message : More than one input Not Allowed

Junit test code:-

Smt. Chandaben Mohanbhai Patel Institute of Computer Applications
Master of Computer Applications (MCA)
Semester - II
CA870 Software Quality Assurance

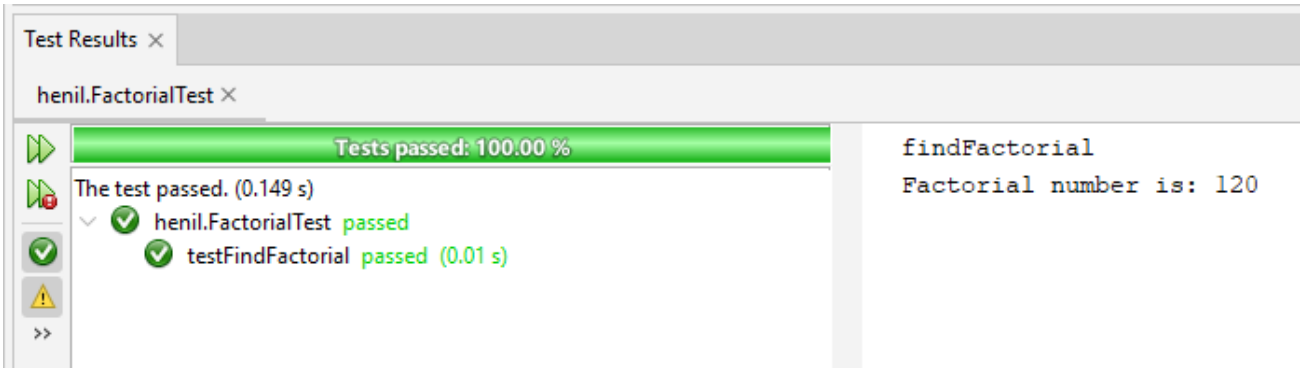
```
package Admin;
public class Factorial {
    public static long findFactorial(int n) {
        if (n == 0 || n == 1)
            return 1;
        else
            return n * findFactorial(n - 1);
    }
}
```

```
package Admin;

import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
import static org.junit.Assert.*;


public class FactorialTest {

    public FactorialTest() {
    }

    @BeforeClass
```

```
public static void setUpClass() {
}

@AfterClass
public static void tearDownClass() {
}

@Before
public void setUp() {
}

@After
public void tearDown() {
}

/**
 * Test of findFactorial method, of class Factorial.
 */
@Test
public void testFindFactorial() {
    System.out.println("findFactorial");
    int n = 5;
    long expResult = 120;
    long result = Factorial.findFactorial(n);
    assertEquals(expResult, result);
    if(expResult==result){
        System.out.println("Factorial number is: "+expResult);
```

```
        }else{
            System.out.println("Expected result is not same");
        }
        // TODO review the generated test code and remove the default call to fail.
       // fail("The test case is a prototype.");
    }

}
```

Output:-



5. Write a program to search a given number froam an array.If number is found,then give the index the number if number is not found return -1.

Check List:

- Any Element of an Input set is Alphabetic
- Any Element of an Input set is Special Symbol
- Any Element of an Input set is Zero
- There are More than two input
- Any Element of an Input set is floating point value.
- First must be array value and second is single value.

List	Checklist	Input	Expected Output

Smt. Chandaben Mohanbhai Patel Institute of Computer Applications
Master of Computer Applications (MCA)
Semester - II
CA870 Software Quality Assurance

1.	Element Of an Input is Alphabetic	A	Error Message: Alphabets Not Allowed
2.	Element Of an Input is Special Symbol	&	Error Message : Special Symbol Not Allowed
3.	Element Of an Input is Zero	0	Error Message : zero Not Allowed
4.	There are More than two input	(15,25,5,10),(16),(5)	Error Message : More than two input Not Allowed
5.	Any Element of an Input set is floating point value.	(2.5,5,5,6),(5)	Error Message : Floating point value is not allowed
6.	First must be array value and second is single value.	(2,5,4.6,),(5)	Error Message:Not change order of sequance

Junit test code:-

```
package Admin;
public class ArraySearch {
    public static int searchNumber(int[] array, int target) {
       for (int i = 0; i < array.length; i++) {
          if (array[i] == target) {
             return i;  // Number found, return the index
          }
       }
       return -1;  // Number not found
    }
}
```

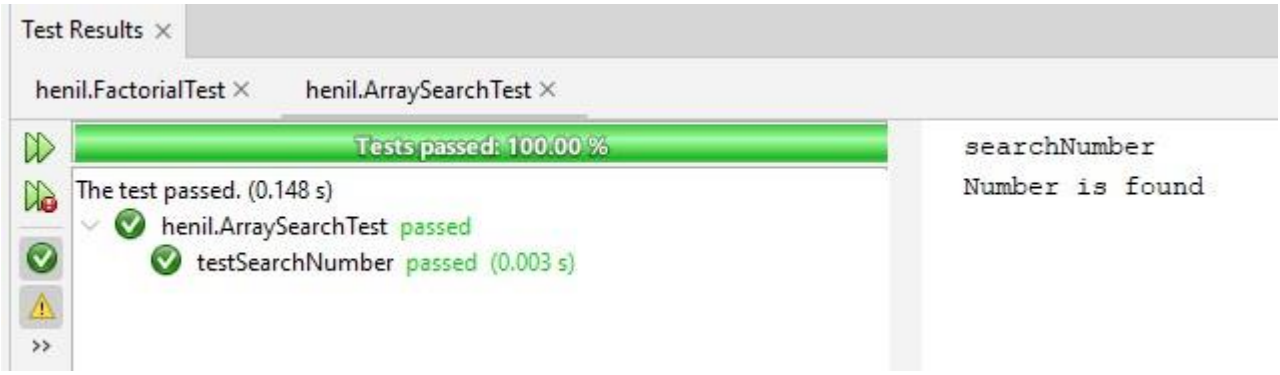
```
package Admin;
```

```
import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
import static org.junit.Assert.*;


public class ArraySearchTest {

    public ArraySearchTest() {
    }

    @BeforeClass
    public static void setUpClass() {
    }

    @AfterClass
    public static void tearDownClass() {
    }

    @Before
    public void setUp() {
    }

    @After
```

```
    public void tearDown() {
    }



    @Test
    public void testSearchNumber() {
        System.out.println("searchNumber");
        int[] array = {2, 5, 8, 12, 16, 20, 25, 30};
        int target = 16;
        int expResult = 4;
        int result = ArraySearch.searchNumber(array, target);
        assertEquals(expResult, result);
        if(expResult==result){
            System.out.println("Number is found ");
        }else{
            System.out.println("Number is not found");
        }
        // TODO review the generated test code and remove the default call to fail.
        //fail("The test case is a prototype.");
    }


}
```

Output:-

Smt. Chandaben Mohanbhai Patel Institute of Computer Applications
Master of Computer Applications (MCA)
Semester - II
CA870 Software Quality Assurance



6. Write a program to find the average values from a given array falling in a given range specified by MIN and MAX.

Check List:

- Any Element of an Input set is Alphabetic
- Any Element of an Input set is Special Symbol
- Any Element of an Input set is Zero
- There are less than three input
- There are More than three input
- Any Element of an Input set is floating point value.
- First must be array value and second is single value.

List	Checklist	Input	Expected Output
1.	Element Of an Input is Alphabetic	A	Error Message: Alphabets Not Allowed
2.	Element Of an Input is Special Symbol	&	Error Message : Special Symbol Not Allowed
3.	Element Of an Input is Zero	0	Error Message : zero Not Allowed
4.	There are More than three input	(15,25,5,10),(16),(5)	Error Message : More than three input Not Allowed
5.	Any Element of an Input set is floating point value.	(2.5,5,5,6),(5)	Error Message : Floating point value is not allowed

6.	First must be array value and second is single value.	(2,5,4.6,),(5)	Error Message:Not change order of sequance
7.	There are less than three input	(2,5,4,8),(10)	Error Message:Less than three input is not valid

Junit test code:-

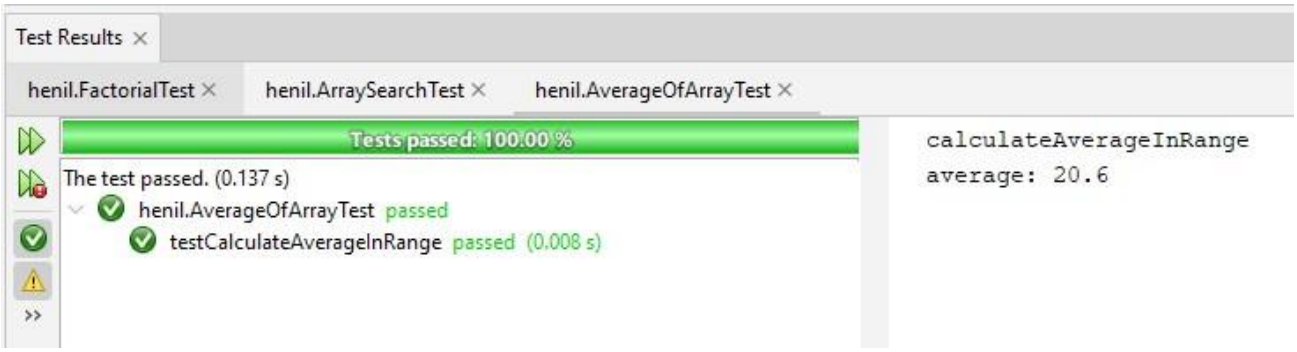
```
package Admin;



public class AverageOfArray {
    public static double calculateAverageInRange(int[] array, int minRange, int maxRange) {
        int sum = 0;
        int count = 0;

        for (int num : array) {
            if (num >= minRange && num <= maxRange) {
                sum += num;
                count++;
            }
        }

        if (count > 0) {
            return (double) sum / count;
        } else {
            return -1;  // No numbers found in the specified range
        }
    }
```

```
}

package Admin;

import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
import static org.junit.Assert.*;


public class AverageOfArrayTest {

    public AverageOfArrayTest() {
    }

    @BeforeClass
    public static void setUpClass() {
    }

    @AfterClass
    public static void tearDownClass() {
    }

    @Before
    public void setUp() {
```

```
    }

    @After
    public void tearDown() {
    }

    /**
     * Test of calculateAverageInRange method, of class AverageOfArray.
     */
    @Test
    public void testCalculateAverageInRange() {
        System.out.println("calculateAverageInRange");
        int[] array = {12, 25, 8, 30, 16, 20, 5, 40};
        int minRange = 10;
        int maxRange = 30;
        double expResult = 20.6;
        double result = AverageOfArray.calculateAverageInRange(array, minRange,
maxRange);
        assertEquals(expResult, result, 0);
         if(expResult==result){
            System.out.println("average:  "+expResult);
        }else{
            System.out.println("Average is not found");
        }
        // TODO review the generated test code and remove the default call to fail.
       // fail("The test case is a prototype.");
    }
```

}

Output:-



7. Write a program to find total number of digits,alphabets and special characters from a given string.

Check List:

- There are More than one input

List	Checklist	Input	Expected Output
1.	There are More than one input	(“abc12$”),(“asd23$”)	Error Message: More than one input not allowed

Junit test code:-

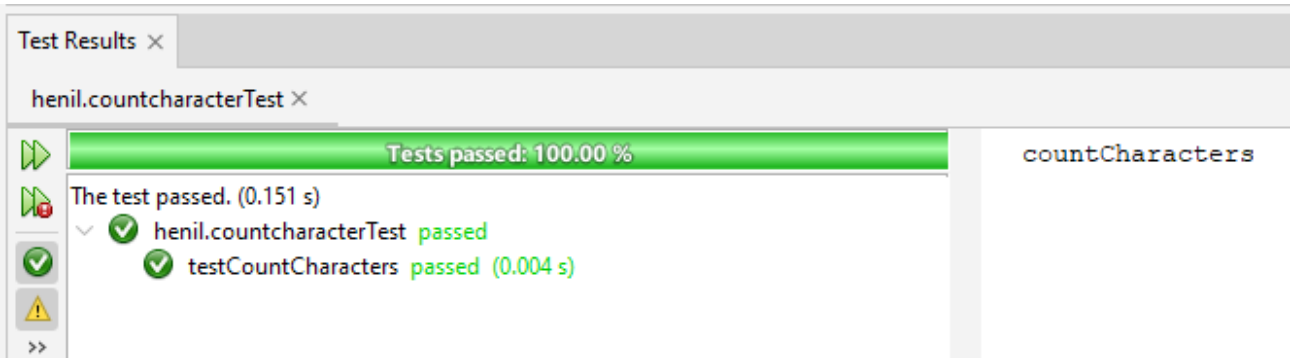
```
package Admin;



public class countcharacter {
  public static int[] countCharacters(String inputString) {
      int digitCount = 0;
      int alphabetCount = 0;
```

```
    int specialCharCount = 0;



    // Iterate through each character in the string
    for (int i = 0; i < inputString.length(); i++) {
        char ch = inputString.charAt(i);

        // Check if the character is a digit
        if (Character.isDigit(ch)) {
            digitCount++;
        }
        // Check if the character is an alphabet
        else if (Character.isLetter(ch)) {
            alphabetCount++;
        }
        // If it is neither digit nor alphabet, consider it as a special character
        else {
            specialCharCount++;
        }
    }



   int[] ans=new int[]{digitCount,alphabetCount,specialCharCount};

    return ans;
}
```

```
}

package Admin;

import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
import static org.junit.Assert.*;



public class countcharacterTest {

    public countcharacterTest() {
    }

    @BeforeClass
    public static void setUpClass() {
    }

    @AfterClass
    public static void tearDownClass() {
    }

    @Before
    public void setUp() {
```

```
    }

    @After
    public void tearDown() {
    }



    @Test
    public void testCountCharacters() {
        System.out.println("countCharacters");
        String inputString = "Admin022Neel";
        int[] expResult = {3,10,0};
        int[] result = countcharacter.countCharacters(inputString);
        assertArrayEquals(expResult, result);

        // TODO review the generated test code and remove the default call to fail.
        //fail("The test case is a prototype.");
    }

}
```

output:-



Smt. Chandaben Mohanbhai Patel Institute of Computer Applications
Master of Computer Applications (MCA)
Semester - II
CA870 Software Quality Assurance

8. Write a program to accept a string and character ,count occurrence of character if found else return -1.

Check List:

- There are More than two input
- Less than two input
- input is not a specialcharacter
- input is not numeric
- order must be follow

List	CheckList	Input	Expected Output
1	There are More than two input	('hello','h','k')	Error:More than two input not allowed
2	Less than two input	('hello')	Error:Less than two input not allowed
3	Input Is not a specialcharacter	('$hello','$')	Error:Special character not allowed
4	Input is not numeric	('34hello','3')	Error:numeric input not allowed
5	Order must be follow	('h','hello')	Error:unorder input not allowed

Junit test code:-

```
package Admin;
```

Smt. Chandaben Mohanbhai Patel Institute of Computer Applications
Master of Computer Applications (MCA)
Semester - II
CA870 Software Quality Assurance

```
public class stringoccur {

    public static int countCharacterOccurrence(String inputString, char searchChar) {

        int count = 0;


        // Iterate through each character in the string

        for (int i = 0; i < inputString.length(); i++) {

            char ch = inputString.charAt(i);


            // Check if the character matches the search character

            if (ch == searchChar) {

                count++;

            }

        }


        // Return the count or -1 if the character is not found

        return count > 0 ? count : -1;

    }

}
```

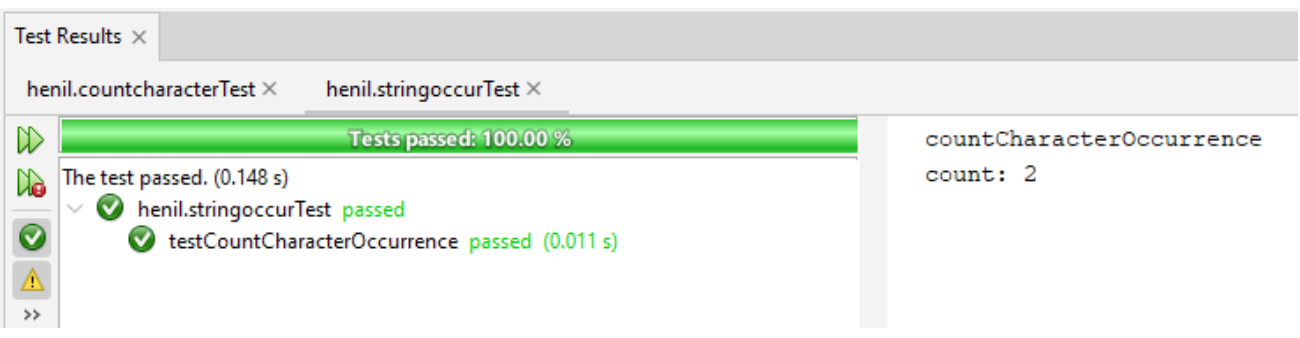
```
package Admin;


import org.junit.After;

import org.junit.AfterClass;

import org.junit.Before;
```

Smt. Chandaben Mohanbhai Patel Institute of Computer Applications
Master of Computer Applications (MCA)
Semester - II
CA870 Software Quality Assurance

```
import org.junit.BeforeClass;

import org.junit.Test;

import static org.junit.Assert.*;



public class stringoccurTest {


    public stringoccurTest() {

    }


    @BeforeClass

    public static void setUpClass() {

    }


    @AfterClass

    public static void tearDownClass() {

    }


    @Before

    public void setUp() {

    }


    @After

    public void tearDown() {
```

```
    }



    @Test
    public void testCountCharacterOccurrence() {
        System.out.println("countCharacterOccurrence");
        String inputString = "Jethalal Tendulkar";
        char searchChar = 'l';
        int expResult = 2;
        int result = stringoccur.countCharacterOccurrence(inputString, searchChar);
        assertEquals(expResult, result);
        if(expResult==result){
            System.out.println("count: "+expResult);
        }else{
            System.out.println("character if not found");
        }
        // TODO review the generated test code and remove the default call to fail.
        //fail("The test case is a prototype.");
    }


}
```

output:-



Smt. Chandaben Mohanbhai Patel Institute of Computer Applications
Master of Computer Applications (MCA)
Semester - II
CA870 Software Quality Assurance

9. Write a program to return count of elements which are divisible by 4,6,7,8 from an array of 10 elemetns.

Check List:

- There are More than one input
- input is an array.
- Not allowed Alphabetic value.
- Any element is an Special character.

List	CheckList	Input	Expected Output
1	There are More than one input	[1,2,3,4],[1,5,9]	Error:More than one input not allowed
2	Input is an array	25	Error:must be in array format
3	Not allowed alphabetic value	[hello,c]	Error:alphabetic value not allowed
4	Any element is an Special character	[$,%,4]	Error:Special character not allowed

Junit test code:-

```
package Admin;



public class divigiblenumber {

    public static int countDivisibleElements(int[] array) {

      int count = 0;
```

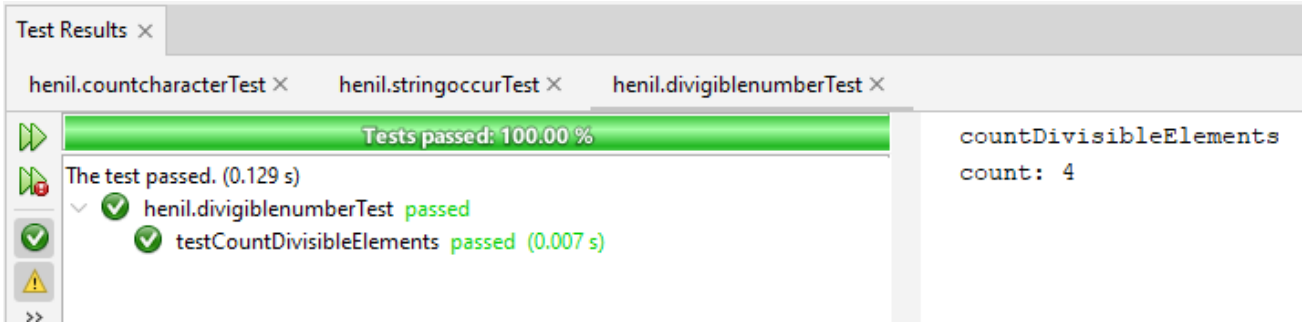
```
        // Iterate through each element in the array
        for (int element : array) {
            // Check if the element is divisible by 4, 6, 7, or 8
            if (element % 4 == 0 || element % 6 == 0 || element % 7 == 0 || element % 8 == 0) {
                count++;
            }
        }

        // Return the count of divisible elements
        return count;
    }
}

package Admin;

import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
import static org.junit.Assert.*;


public class divigiblenumberTest {

    public divigiblenumberTest() {
    }
```

Smt. Chandaben Mohanbhai Patel Institute of Computer Applications
Master of Computer Applications (MCA)
Semester - II
CA870 Software Quality Assurance

```
@BeforeClass
public static void setUpClass() {
}

@AfterClass
public static void tearDownClass() {
}

@Before
public void setUp() {
}

@After
public void tearDown() {
}



@Test
public void testCountDivisibleElements() {
   System.out.println("countDivisibleElements");
   int[] array = {5,88,75,65,30,21,7,45,50,10};
   int expResult = 4;
   int result = divigiblenumber.countDivisibleElements(array);
   assertEquals(expResult, result);
   if(expResult==result){
      System.out.println("count: "+expResult);
```

```
    }else{
       System.out.println("number is not divigible by 4,6,7,8");
    }
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
  }

}
```

output:-



10. Write a program that will count total number of occurences of given character from a string.

Check List:

- There are More than two input
- Less than two input
- input is not a specialcharacter
- input is not numeric
- order must be follow

List	CheckList	Input	Expected Output

Smt. Chandaben Mohanbhai Patel Institute of Computer Applications
Master of Computer Applications (MCA)
Semester - II
CA870 Software Quality Assurance

1	There are More than two input	('hello','h','k')	Error:More than two input not allowed
2	Less than two input	('hello')	Error:Less than two input not allowed
3	Input Is not a specialcharacter	('$hello','$')	Error:Special character not allowed
4	Input is not numeric	('34hello','3')	Error:numeric input not allowed
5	Order must be follow	('h','hello')	Error:unorder input not allowed

Junit test code:-

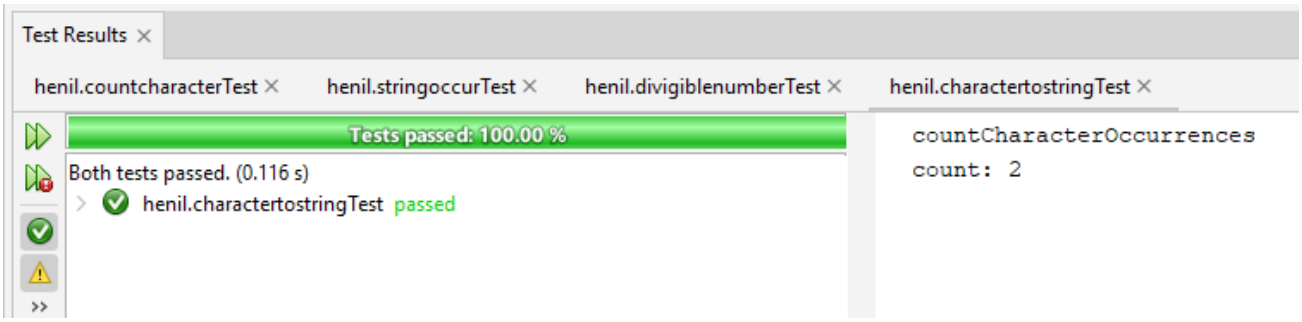
```
package Admin;



public class charactertostring {
    public static int countCharacterOccurrences(String inputString, char searchChar) {
        int count = 0;

        // Iterate through each character in the string
        for (int i = 0; i < inputString.length(); i++) {
            char ch = inputString.charAt(i);

            // Check if the character matches the search character
            if (ch == searchChar) {
                count++;
            }
```

```
        }

        // Return the count of occurrences
        return count;
    }
}

package Admin;

import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
import static org.junit.Assert.*;


public class charactertostringTest {

    public charactertostringTest() {
    }

    @BeforeClass
    public static void setUpClass() {
    }

    @AfterClass
```

```
public static void tearDownClass() {
}

@Before
public void setUp() {
}

@After
public void tearDown() {
}

@Test
public void testSomeMethod() {
    // TODO review the generated test code and remove the default call to fail.
    //fail("The test case is a prototype.");
}


@Test
public void testCountCharacterOccurrences() {
    System.out.println("countCharacterOccurrences");
    String inputString = " Jethalal Tendulkar ";
    char searchChar = 'l';
    int expResult = 2;
    int result = charactertostring.countCharacterOccurrences(inputString, searchChar);
    assertEquals(expResult, result);
    if(expResult==result){
```

```
        System.out.println("count: "+expResult);
    }else{
        System.out.println("charcater is not found");
    }
    // TODO review the generated test code and remove the default call to fail.
    // fail("The test case is a prototype.");
  }

}
```

output:-



11. Write a program to convert a string into reverse and print the result string.

CheckList:-

- More than one input.
- Input Numeric value
- input special character

List	CheckList	Input	Expected Output
1	There are More than one input	('hello','abc')	Error:More than one input not allowed

2	Input numeric value	-123	Error:numeric value is not allowed
3	Input specialcharacter	('$')	Error:Special character not allowed

Junit test code :-

```
package Admin;



public class reversestring {
   public static String reverseString(String inputString) {
      // Convert the string to a character array
      char[] charArray = inputString.toCharArray();

      // Reverse the character array
      for (int i = 0, j = charArray.length - 1; i < j; i++, j--) {
         char temp = charArray[i];
         charArray[i] = charArray[j];
         charArray[j] = temp;
      }

      // Convert the character array back to a string
      return new String(charArray);
   }
}
```

Smt. Chandaben Mohanbhai Patel Institute of Computer Applications
Master of Computer Applications (MCA)
Semester - II
CA870 Software Quality Assurance

```
package Admin;


import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
import static org.junit.Assert.*;


public class reversestringTest {


   public reversestringTest() {
   }


   @BeforeClass
   public static void setUpClass() {
   }
   @AfterClass
   public static void tearDownClass() {
   }
   @Before
   public void setUp() {
   }
   @After
   public void tearDown() {
   }
   @Test
```

Smt. Chandaben Mohanbhai Patel Institute of Computer Applications
Master of Computer Applications (MCA)
Semester - II
CA870 Software Quality Assurance

```
    public void testReverseString() {
        System.out.println("reverseString");
        String inputString = "Admin";
        String expResult = "lineH";
        String result = reversestring.reverseString(inputString);
        assertEquals(expResult, result);
        if(expResult.equals(result)){
            System.out.println("reverse string is: "+expResult);
        }else{
            System.out.println("string not found");
        }
        // TODO review the generated test code and remove the default call to fail.
       // fail("The test case is a prototype.");
    }
}
```

output:-