Assignment on Jumble

Submitted by: -

MTech, Software Engineering

Sub – Software Testing Lab

National Institute of Technology

Rourkela

1. **Write a program to generate Factorial of number (where stack length should be at 3 (max)). The numbers should be 5, 3, 8 and 15.**

Package Explorer | JUnit

Finished after 0.044 seconds

Runs: 2/2 Errors: 0 Failures: 0

FactorialTest [Runner: JUnit 3] (0.003 s)

ArraySum.java | ArraySumTest.java | OddEven.java | OddEvenTest.java | Factorial.ja

```java
import junit.framework.TestCase;

public class FactorialTest extends TestCase {
    public void test1()
    {

        assertEquals("120",Factorial.factorial(5));
    }

    public void test2()
    {

        assertEquals("factorial doesn't exist",Factorial.factorial(-1));
    }

```

ArraySum.java | ArraySumTest.java | OddEven.java | OddEvenTest.java

```java

public class Factorial {
    public static String factorial(int n)
    {
        if(n<0)
            return "factorial doesn't exist";
        else
        {
            long[] fact = new long[n+1];
            fact[0] = 1;
            for(int i=1;i<=n;i++)
            {
                fact[i]= fact[i-1]*i;
            }
            return ""+fact[n];
        }
    }

}

```

Problems | Javadoc | Declaration | Console

```
<terminated> Run Jumble [Java Application] C:\Program Files\Java\jre1.8.0_202\bin\javaw.exe
Mutating Factorial
Tests: FactorialTest
Mutation points = 14, unit test time limit 2.01s
...............
Jumbling took 7.47s
Score: 100%
```

2. Write a program to generate Fibonacci numbers.

Package Explorer | JUnit

Finished after 0.025 seconds

Runs: 3/3 Errors: 0 Failures: 0

FibonacciTest [Runner: JUnit 4] (0.001 s)

FibonacciTest.java | Fibonacci.java

```java
import junit.framework.TestCase;

public class FibonacciTest extends TestCase {

    public void test1() {
        assertEquals("0",Fibonacci.fibonacci(1));
    }

    public void test2() {
        assertEquals("0 1 1",Fibonacci.fibonacci(3));
    }

    public void test3() {
        assertEquals("0 1 1 2 3",Fibonacci.fibonacci(5));
    }

}
```

FibonacciTest.java | Fibonacci.java

```java
public class Fibonacci {
    public static String fibonacci(int n) {
        int f1=0,f2=1,f3;
        String p = null;
        if(n==1) {
            return "0";
        }
        else if(n==2) {
            return "0 1";
        }
        else {
            p = "0 1";
            for(int i=2;i<n;++i)
            {
                f3=f1+f2;
                p = p+" "+f3;
                f1=f2;
                f2=f3;
            }
            return p;
```

Problems | Javadoc | Declaration | Console

```
<terminated> Run Jumble [Java Application] C:\Program Files\Java\jre1.8.0_202\bin\javaw.exe
Mutating Fibonacci
Tests: FibonacciTest
Mutation points = 16, unit test time limit 2.01s
.............T..
Jumbling took 11.374s
Score: 100%
```

3. **Write a program which performs sorting of group of integer values using quick sort technique.**

Package Explorer | JUnit

Finished after 0.028 seconds

Runs: 3/3 Errors: 0 Failures: 0

Quick_SortTest [Runner: JUnit 4] (0.000 s)

Quick_Sort.java | Quick_SortTest.java

```java
import junit.framework.TestCase;

public class Quick_SortTest extends TestCase {

    public void test1() {
        int arr[] = {12,45,7,5};
        assertEquals("5,7,12,45,",Quick_Sort.quick_sort(arr));
    }

    public void test2() {
        int arr[] = {8,7,5,6,2,3,1,4,5,4};
        assertEquals("1,2,3,4,4,5,5,6,7,8,",Quick_Sort.quick_sort(arr));
    }

    public void test3() {
        int arr[] = {12,11,13,5,6,7};
        assertEquals("5,6,7,11,12,13,",Quick_Sort.quick_sort(arr));
    }

```

Quick_Sort.java | Quick_SortTest.java

```java
public class Quick_Sort {
    public static int partition(int arr[], int low, int high)
    {
        int pivot = arr[high];
        int i = (low-1);
        for (int j=low; j<high; j++)
        {
            if (arr[j] <= pivot)
            {
                i++;
                int temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }

        int temp = arr[i+1];
        arr[i+1] = arr[high];
        arr[high] = temp;

```

Problems | Javadoc | Declaration | Console

```
<terminated> Run Jumble [Java Application] C:\Program Files\Java\jre1.8.0_202\bin\javaw.exe
Mutating Quick_Sort
Tests: Quick_SortTest
Mutation points = 27, unit test time limit 2.02s
..................M FAIL: (Quick_Sort.java:30): 1 -> 0
........
Jumbling took 13.141s
Score: 96%
```

4. **Write a program that accepts elements of a matrix and display its transpose.**

Package Explorer | JUnit

Finished after 0.022 seconds

Runs: 3/3 Errors: 0 Failures: 0

Transpose_MatrixTest [Runner: JUnit 4] (0.001 s)

Transpose_Matrix.java | Transpose_MatrixTest.java

```java
import junit.framework.TestCase;

public class Transpose_MatrixTest extends TestCase {

    public void test1() {
        int arr[][]={{1,3,4},{2,4,3},{3,4,5}};
        assertEquals("1 2 3 ||3 4 4 ||4 3 5 ||",Transpose_Matrix.transpose_matrix(arr));
    }

    public void test2() {
        int arr[][]={{1,2,3},{4,5,6},{7,8,9}};
        assertEquals("1 4 7 ||2 5 8 ||3 6 9 ||",Transpose_Matrix.transpose_matrix(arr));
    }

    public void test3() {
        int arr[][]={{9,1,4},{6,3,7},{4,4,3}};
        assertEquals("9 6 4 ||1 3 4 ||4 7 3 ||",Transpose_Matrix.transpose_matrix(arr));
    }

}
```

Transpose_Matrix.java | Transpose_MatrixTest.java

```java
public class Transpose_Matrix {
    public static String transpose_matrix(int arr[][]) {

        int transpose[][]=new int[3][3];

        for(int i=0;i<3;i++){
            for(int j=0;j<3;j++){
                transpose[i][j]=arr[j][i];
            }
        }
        String p = "";
        for(int i=0;i<3;i++){
            for(int j=0;j<3;j++){
                p = p+transpose[i][j]+" ";
            }
            p = p+"||";
        }
        return p;
    }
}
```

Problems | Javadoc | Declaration | Console

```
<terminated> Run Jumble [Java Application] C:\Program Files\Java\jre1.8.0_202\bin\javaw.exe
Mutating Transpose_Matrix
Tests: Transpose_MatrixTest
Mutation points = 22, unit test time limit 2.01s
...M FAIL: (Transpose_Matrix.java:4): 3 -> 4
M FAIL: (Transpose_Matrix.java:4): 3 -> 4
.................
Jumbling took 11.108s
Score: 90%
```

5. Write a program to add two matrices and display sum matrix.

Package Explorer | JUnit

Finished after 0.025 seconds

Runs: 3/3 Errors: 0 Failures: 0

MatrixAdditionTest [Runner: JUnit 4] (0.000 s)

Transpose_MatrixTest.java | MatrixAddition.java | MatrixAdditionTest.java

```java
import junit.framework.TestCase;

public class MatrixAdditionTest extends TestCase {

    public void test1() {
        int a[][]={{1,4,4},{2,4,4},{4,4,5}};
        int b[][]={{2,4,4},{2,4,4},{1,2,4}};
        assertEquals("3 8 8 ||4 8 8 ||5 6 9 ||",MatrixAddition.matrixaddition(a,b));
    }

    public void test2() {
        int a[][]={{4,4,1},{2,6,3},{4,7,5}};
        int b[][]={{4,4,7},{4,4,8},{1,0,4}};
        assertEquals("8 8 8 ||6 10 11 ||5 7 9 ||",MatrixAddition.matrixaddition(a,b));
    }

    public void test3() {
        int a[][]={{1,6,4},{4,9,2},{1,9,2}};
        int b[][]={{1,9,2},{4,4,8},{1,6,4}};
        assertEquals("2 15 6 ||8 13 10 ||2 15 6 ||",MatrixAddition.matrixaddition(a,b));
```

MatrixAddition.java | MatrixAdditionTest.java

```java
public class MatrixAddition {
    public static String matrixaddition(int a[][],int b[][]) {

        int c[][]=new int[3][3];
        String p = "";
        for(int i=0;i<3;i++){
            for(int j=0;j<3;j++){
                c[i][j]=a[i][j]+b[i][j];
                p = p+c[i][j]+" ";
            }
            p = p+"||";
        }
        return p;
    }
}
```

Problems | Javadoc | Declaration | Console

```
<terminated> Run Jumble [Java Application] C:\Program Files\Java\jre1.8.0_202\bin\javaw.exe
Mutating MatrixAddition
Tests: MatrixAdditionTest
Mutation points = 15, unit test time limit 2.01s
...M FAIL: (MatrixAddition.java:4): 3 -> 4
M FAIL: (MatrixAddition.java:4): 3 -> 4
..........
Jumbling took 8.013s
Score: 86%
```

6. **Write a program to Print Prime Numbers from 1 to 100 using Scanner Class and For Loop.**

Package Explorer | JUnit

Finished after 0.022 seconds

Runs: 1/1 Errors: 0 Failures: 0

PrimeTest [Runner: JUnit 4] (0.000 s)

Prime.java | PrimeTest.java

```java
import junit.framework.TestCase;

public class PrimeTest extends TestCase {

    public void test1() {
        assertEquals("2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 ",Prime.prime());
    }

}
```

Prime.java | PrimeTest.java

```java
    public static boolean isprime(int n) {
        int i,m=0,flag=0;
          m=n/2;
          if(n==0||n==1){
           return false;
          }
          else{
           for(i=2;i<=m;i++){
            if(n%i==0){
             flag=1;
             break;
            }
           }
           if(flag==0){
               return true;
           }
           if(flag==1){
               return false;
           }

          }
```

Problems | Javadoc | Declaration | Console

```
<terminated> Run Jumble [Java Application] C:\Program Files\Java\jre1.8.0_202\bin\jav
Mutating Prime
Tests: PrimeTest
Mutation points = 32, unit test time limit 2.01s
..M FAIL: (Prime.java:3): 0 -> 1
................M FAIL: (Prime.java:18): 1 -> 0
M FAIL: (Prime.java:18): negated conditional
..M FAIL: (Prime.java:23): 0 -> 1
.M FAIL: (Prime.java:27): 1 -> 0
.T...
Jumbling took 18.457s
Score: 84%
```

7. Write a program to generate palindrome of numbers.

Package Explorer | JUnit

Finished after 0.034 seconds

Runs: 1/1 Errors: 0 Failures: 0

PalindromeTest [Runner: JUnit 4] (0.000 s)

Prime.java | PrimeTest.java | Palindrome.java | PalindromeTest.java

```java
import junit.framework.TestCase;
public class PalindromeTest extends TestCase{
    public void test1()
    {

        assertEquals("Palindromes are: 22",Palindrome.check(22,30));
    }

}

```

Prime.java | PrimeTest.java | Palindrome.java | PalindromeTest.java

```java

public class Palindrome {
    public static String check(int n1,int n2)
    {
        String palins="Palindromes are:";
        for(int i=n1;i<=n2;i++)
        {
            StringBuffer rev= new StringBuffer(""+i);
            rev.reverse();
            String rev_t= rev.toString();
            if(rev_t.equals(""+i))
            {
                palins=palins+" "+rev;
            }
        }
        return palins;
    }

}

```

Problems | Javadoc | Declaration | Console

```
<terminated> Run Jumble [Java Application] C:\Program Files\Java\jre1.8.0_202\bin\javaw.exe
Mutating Palindrome
Tests: PalindromeTest
Mutation points = 6, unit test time limit 2.01s
...T..
Jumbling took 5.5s
Score: 100%
```

8. Write a program to find out sum of two array.

Package Explorer | JUnit

Finished after 0.02 seconds

Runs: 2/2 Errors: 0 Failures: 0

ArraySumTest [Runner: JUnit 4] (0.000 s)

ArraySum.java | ArraySumTest.java

```java
import junit.framework.TestCase;

public class ArraySumTest extends TestCase {

    public void test1() {
        int a[] = {2,3,7,6,8};
        int b[] = {3,8,21,7,9};
        assertEquals("74",ArraySum.array_sum(a,b));
    }

    public void test2() {
        int a[] = {1,2,3};
        int b[] = {3,2,1};
        assertEquals("12",ArraySum.array_sum(a,b));
    }

}
```

ArraySum.java | ArraySumTest.java

```java
public class ArraySum {
    public static String array_sum(int a[],int b[]) {
        int sum = 0;
        for(int i=0;i<a.length;i++) {
            sum = sum + a[i] + b[i];
        }
        return ""+sum;
    }
}
```

Problems | Javadoc | Declaration | Console

```
<terminated> Run Jumble [Java Application] C:\Program Files\Java\jre1.8.0_202\bin\javav
Mutating ArraySum
Tests: ArraySumTest
Mutation points = 7, unit test time limit 2.01s
.......
Jumbling took 4.263s
Score: 100%
```

9. Write a program to check whether the number is even or odd.

Package Explorer | JUnit

Finished after 0.02 seconds

Runs: 2/2 Errors: 0 Failures: 0

OddEvenTest [Runner: JUnit 4] (0.001 s)

OddEven.java | OddEvenTest.java

```java
import junit.framework.TestCase;

public class OddEvenTest extends TestCase {

    public void test1() {
        assertEquals("Even",OddEven.odd_even(44));
    }

    public void test2() {
        assertEquals("Odd",OddEven.odd_even(77));
    }

}
```

OddEven.java | OddEvenTest.java

```java
public class OddEven {
    public static String odd_even(int n) {
        if(n%2==0) {
            return "Even";
        }
        else {
            return "Odd";
        }
    }
}
```

Problems | Javadoc | Declaration | Console

```
<terminated> Run Jumble [Java Application] C:\Program Files\Java\jre1.8.0_202\bin\javaw.exe
Mutating OddEven
Tests: OddEvenTest
Mutation points = 7, unit test time limit 2.0s
.......
Jumbling took 3.834s
Score: 100%
```

10. Write a program for binary to hexa-decimal conversion.

Package Explorer | JUnit

Finished after 0.029 seconds

Runs: 3/3 Errors: 0 Failures: 0

Bin_to_hexaTest [Runner: JUnit 4] (0.000 s)

Bin_to_hexa.java | Bin_to_hexaTest.java

```java
import junit.framework.TestCase;

public class Bin_to_hexaTest extends TestCase {

    public void test1() {
        assertEquals("A",Bin_to_hexa.dec2hex(1010));
    }

    public void test2() {
        assertEquals("26",Bin_to_hexa.dec2hex(100110));
    }

    public void test3() {
        assertEquals("3B",Bin_to_hexa.dec2hex(111011));
    }

}
```

Bin_to_hexa.java | Bin_to_hexaTest.java

```java
public class Bin_to_hexa {
    public static int bin_to_dec(long binary)
    {
        int decimal = 0, i = 0;

        while (binary > 0) {
            decimal+= Math.pow(2, i++) * (binary % 10);
            binary /= 10;
        }
        return decimal;
    }

    public static String dec2hex(long binary)
    {
        int decimalNumber = bin_to_dec(binary);
        String hex = Integer.toHexString(decimalNumber);
        hex = hex.toUpperCase();
        return hex;
    }
}
```

Problems | Javadoc | Declaration | Console

<terminated> Run Jumble [Java Application] C:\Program Files\Java\jre1.8.0_202\bin\javaw.exe

```
Mutating Bin_to_hexa
Tests: Bin_to_hexaTest
Mutation points = 13, unit test time limit 2.01s
.............
Jumbling took 7.088s
Score: 100%
```