Day 2 : Special Logic Building Assignment: 10 special Recursion Programs

1. Print 1 to n without using loops

Output:

12345678910

2. Sum of natural numbers using recursion

```
package Assig2;

public class Q2{

   public static int sum(int n) {
      if (n == 1) {
        return 1;
      } else {
```

```
return n + sum(n - 1);
}

public static void main(String[] args) {
    int n = 10;
    int sum = sum(n);
    System.out.println("Sum of natural numbers from 1 to " + n + " is:
" + sum);
    }
}
```

Sum of natural numbers from 1 to 10 is: 55

3. Mean of Array using Recursion

```
package Assig2;
public class Q3 {
    public static int arraySum(int[] arr, int index) {
        if (index == arr.length - 1) {
           return arr[index];
        } else {
           return arr[index] + arraySum(arr, index + 1);
    }
    public static double arrayMean(int[] arr, int index) {
        if (index == arr.length - 1) {
            return (double) arr[index] / arr.length;
            return ((double) arr[index] + arraySum(arr, index + 1)) /
arr.length;
       }
    }
    public static void main(String[] args) {
        int[] array = {1, 2, 3, 4, 5};
        double mean = arrayMean(array, 0);
        System.out.println("Mean of the array is: " + mean);
    }
}
```

```
Mean of the array is: 3.0
```

4. Sum of array elements using recursion

```
package Assig2;

public class Q4 {

    public static int arraySum(int[] arr, int index) {
        if (index == arr.length - 1) {
            return arr[index];
        } else {
            return arr[index] + arraySum(arr, index + 1);
        }

    public static void main(String[] args) {
        int[] array = {1, 2, 3, 4, 5};
        int sum = arraySum(array, 0);
        System.out.println("Sum of the array elements is: " + sum);
    }
}
```

Output:

```
Sum of the array elements is: 15
```

5. Decimal to binary number using recursion

```
package Assig2;
public class Q5{
```

```
public static String decimalToBinary(int decimal) {
    if (decimal == 0) {
        return "0";
    } else if (decimal == 1) {
        return "1";
    } else {
        return decimalToBinary(decimal / 2) + decimal % 2;
    }
}

public static void main(String[] args) {
    int decimal = 10;
    String binary = decimalToBinary(decimal);
    System.out.println("Binary representation of " + decimal + " is: " + binary);
    }
}
```

Binary representation of 10 is: 1010

6. Sum of digit of a number using recursion

```
package Assig2;
public class Q6 {

   public static int sumOfDigits(int number) {
        if (number == 0) {
            return 0;
        } else {
            return number % 10 + sumOfDigits(number / 10);
        }
   }

   public static void main(String[] args) {
        int number = 12345;
        int sum = sumOfDigits(number);
        System.out.println("Sum of digits of " + number + " is: " + sum);
   }
}
```

```
Sum of digits of 12345 is: 15
```

7. Print reverse of a string using recursion

```
package Assig2;
public class Q7 {

   public static void printReverse(String str, int index) {
      if (index < str.length()) {
           printReverse(str, index + 1);
           System.out.print(str.charAt(index));
      }
   }

   public static void main(String[] args) {
      String str = "hello";
        printReverse(str, 0);
   }
}</pre>
```

Output:

Olleh

8. Program for length of a string using recursion

```
package Assig2;
public class Q8 {
    public static int stringLength(String str) {
        if (str.equals("")) {
            return 0;
        } else {
```

```
return 1 + stringLength(str.substring(1));
}

public static void main(String[] args) {
    String str = "hello";
    int length = stringLength(str);
    System.out.println("Length of the string \"" + str + "\" is: " + length);
}
```

Length of the string "hello" is: 5

9. Tail recursion to calculate sum of array elements.

```
package Assig2;
public class Q9{

   public static int arraySum(int[] arr, int index, int sumSoFar) {
        if (index == arr.length) {
            return sumSoFar;
        } else {
            return arraySum(arr, index + 1, sumSoFar + arr[index]);
        }

   public static void main(String[] args) {
        int[] array = {1, 2, 3, 4, 5};
        int sum = arraySum(array, 0, 0);
        System.out.println("Sum of the array elements is: " + sum);
   }
}
```

Output:

```
Sum of the array elements is: 15
```

10. Recursive function to check if a string is palindrome

```
package Assig2;
public class Q10{
    public static boolean isPalindrome(String str) {
        if (str.length() <= 1) {
            return true;
        return str.charAt(0) == str.charAt(str.length() - 1)
               && isPalindrome(str.substring(1, str.length() - 1));
    public static void main(String[] args) {
        String str = "radar";
        if (isPalindrome(str)) {
            System.out.println(str + " is a palindrome.");
        } else {
            System.out.println(str + " is not a palindrome.");
        }
    }
}
Output:
radar is a palindrome.
```

11. Print Fibonacci Series in reverse order using Recursion

```
package Assig2;

public class Q11 {

    public static int fibonacci(int n) {
        if (n <= 1) {
            return n;
        } else {
            return fibonacci(n - 1) + fibonacci(n - 2);
        }

    public static void printReverseFibonacci(int n) {
        if (n <= 0) {
            return;
        } else {
            System.out.print(fibonacci(n) + " ");
            printReverseFibonacci(n - 1);
        }
}</pre>
```

```
public static void main(String[] args) {
   int count = 10;
   System.out.println("Fibonacci series in reverse order:");
   printReverseFibonacci(count);
}
```

```
Fibonacci series in reverse order: 55 34 21 13 8 5 3 2 1 1
```