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

1. Print 1 to n without using loops

```
public class Q1
{
    public static void printNNumbers(int n)
    {

        if(n > 0)
        {
            printNNumbers(n-1);
            System.out.print(n +" ");

        }

    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter number : ");
        int n = sc.nextInt();
        printNNumbers(n);
    }
}
/*
    Enter number :
    20
    1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
*/
```

2. Sum of natural numbers using recursion

```
public class Q6
{
    public static int sumNaturalNumbers(int n)
    {

        if(n <= 1)
            return n;
        else
            return n+ sumNaturalNumbers(n-1);

    }
    public static void main(String[] args) {
          Scanner sc = new Scanner(System.in);
          System.out.println("Enter number : ");
          int n = sc.nextInt();
          int sum = sumNaturalNumbers(n);
          System.out.println("Sum = "+sum+" ");
     }
}

     Enter number :
     10
     Sum = 55
*/
```

3. Mean of Array using Recursion

```
public class Q3
{
     public static double meanArrayElement(int arr[],int n)
     {

          if(n == 0)
               return 0.0;
          else
               return (meanArrayElement(arr,n-1)*(n-1) + arr[n-
1]) / n;

     }
     public static void main(String[] args) {
          int arr[] = {1,2,3,4,5,6,7,8,9,10};
          System.out.println("Mean of array elements :
"+meanArrayElement(arr,arr.length));
     }
}
/*
 Mean of array elements : 5.5
*/
```

3. Sum of array elements using recursion

```
public class Q4
{
     public static int  sumArrayElement(int arr[],int n)
     {

          if(n <= 0)
```

```
                return 0;
            else
                return  arr[n-1] + sumArrayElement(arr,n-1);

    }
    public static void main(String[] args) {
        int arr[] = {1,2,3,4,5,6,7,8,9,10};
        System.out.println("sum of array elements :
"+sumArrayElement(arr,arr.length));
    }
}
/*
 sum of array elements : 55
*/
```

4. Decimal to binary number using recursion

```
public class Q5
{
    public static void printBinaryNumbers(int n)
    {

        if(n == 0)
            return;

        printBinaryNumbers(n/2);
        System.out.print((n % 2)+" ");

    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter decimal number : ");
        int n = sc.nextInt();
        printBinaryNumbers(n);
    }
}
/*
    Enter decimal number :
    10
    1 0 1 0
*/
```

5. Sum of digit of a number using recursion

```
public class Q6
{
    public static int  sumOfDigit(int n)
    {
        if(n == 0)
            return 0;
        else
            return  sumOfDigit(n/10)+ n%10;

    }
    public static void main(String[] args) {
        int n = 7894;
        System.out.println("The sum  of digit of given number 
"+n+" : "+sumOfDigit(n));
    }
}
/*
 The sum  of digit of given number 7894 : 28

*/
```

6. Print reverse of a string using recursion

```
public class Q7
{
    public static String reverseStr(String s)
    {
        if(s.length() == 0 || s.length() == 1)
            return s;
        else
            return  reverseStr(s.substring(1))+s.charAt(0);

    }
    public static void main(String[] args) {
        String str = "tnameH";
        System.out.println("reverse String of "+str+" : 
"+reverseStr(str));
    }
}
/*
 reverse String of tnameH : Hemant
*/
```

7. Program for length of a string using recursion

```
public class Q8
{
    public static int lengthStr(String s)
    {
        if(s.length() == 0 )
            return 0;
        else
            return  lengthStr(s.substring(1))+1;

    }
    public static void main(String[] args) {
        String str = "Hemant";
        System.out.println("Length of the String "+str+" : 
"+lengthStr(str));
    }
}
/*
 Length of the String Hemant : 6
*/
```

8. Tail recursion to calculate sum of array elements.

```
import java.util.Scanner;
public class Q9
{
    public static int sumArrayElements(int arr[],int n)
    {
        if(n == 0)
            return 0;
        int sum = sumArrayElements(arr,n-1);
        return sum+arr[n-1];
    }
    public static void main(String[] args) {

        int arr[] = {1,2,3,4,5,6,7,8,9,10};
        System.out.println("Sum of ArrayElements Are : 
"+sumArrayElements(arr,arr.length));

    }
}
/*
    Sum of ArrayElements Are : 55
*/
```

9. Recursive function to check if a string is palindrome

```java
public class Q9
{
    public static boolean isPalindrome(String s)
    {
        if(s.length() == 0 || s.length() == 1)
            return true;
        if(s.charAt(0) != s.charAt(s.length()-1))
            return  false;
        return isPalindrome(s.substring(1,s.length()-1));
    }
    public static void main(String[] args) {
        //String str = "nitin";
        String str = "hemant";
        if(isPalindrome(str))
            System.out.println(str+" is Palindrome");
        else
            System.out.println(str+" is not Palindrome");
    }
}
/*
    nitin is Palindrome

    hemant is not Palindrome

*/
```

10. Print Fibonacci Series in reverse order using Recursion

```java
import java.util.*;
class Q10
{
    public static List<Integer>list=new ArrayList<>();

    public static List<Integer> printFib(int n)
    {
     if(n==1)
     {
         list.add(0);
         return list;
     }
     if(n==2)
     {
         list.add(0);
         list.add(1);
         return list;
     }
     List<Integer>result=printFib(n-1);
     int fLast=result.get(result.size()-1);
     int sLast=result.get(result.size()-2);
     int last=fLast+sLast;

     if(last<n)
     {

      result.add(last);
    }
    return result;

    }

    public  static void printReverse(List<Integer>list)
    {
     if(list.size()==0)return;
     Integer val=list.get(0);
     list.remove(val);
     printReverse(list);
     System.out.print(val+" ");

    }

    public static void main(String[]args)
    {
     Scanner sc=new Scanner(System.in);
     System.out.println("Enter number : ");
     int n=sc.nextInt();
     List<Integer>res=printFib(n);
     printReverse(res);
    }
}
/*
    Enter number :
    10
    8 5 3 2 1 1 0
*/
```