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

1. **Print 1 to n without using loops.**

Ans:

import java.util.Scanner;

public class Program1 {

private static void display(int n) {

if(n>0) {

display(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();

display(n);

}

}

1. **Sum of natural numbers using recursion.**

Ans:

import java.util.Scanner;

public class Program2 {

private static int display(int n) {

if(n<=1)

return n;

return n + display(n-1);

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter number: ");

int n = sc.nextInt();

System.out.println(display(n));

}

}

1. **Mean of Array using Recursion.**

Ans:

import java.util.Scanner;

public class Program3 {

private static float mean(int arr[], int n) {

if(n==1)

return (float)arr[n-1];

else

return ((float)(mean(arr, n-1)\*(n-1)+arr[n-1])/n);

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter no of elements: ");

int n = sc.nextInt();

int arr[] = new int[n];

for(int i=0; i<n; i++) {

arr[i]= sc.nextInt();

}

System.out.println(mean(arr, n));

}

}

1. **Sum of array elements using recursion.**

Ans:

import java.util.Scanner;

public class Progrma4 {

private static int sum(int arr[], int n) {

if(n==1)

return arr[n-1];

else

return (sum(arr, n-1)+arr[n-1]);

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter no of elements: ");

int n = sc.nextInt();

int arr[] = new int[n];

for(int i=0; i<n; i++) {

arr[i]= sc.nextInt();

}

System.out.println(sum(arr, n));

}

}

1. **Decimal to binary number using recursion.**

Ans:

public class Program5 {

private static int display(int n) {

if(n==0)

return 0;

else

return (n%2 +10\*display(n/2));

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter number: ");

int n = sc.nextInt();

System.out.print(display(n));

}

}

1. **Sum of digit of a number using recursion.**

Ans: public class Program6 {

private static int display(int n) {

if(n==0)

return 0;

else

return (n%10 +display(n/10));

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter number: ");

int n = sc.nextInt();

System.out.print(display(n));

}

}

1. **Print reverse of a string using recursion**

Ans:

import java.util.Scanner;

public class Program7 {

public static void display(String str) {

if((str==null)||(str.length()<=1))

System.out.print(str);

else {

System.out.print(str.charAt(str.length()-1));

display(str.substring(0, str.length()-1));

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the string: ");

String str = sc.nextLine();

display(str);

sc.close();

}

}

1. Program for length of a string using recursion.

Ans: import java.util.Scanner;

public class Program8 {

private static int length(String str) {

if(str.equals(""))

return 0;

else

return length(str.substring(1))+1;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the string: ");

String str = sc.nextLine();

System.out.println(length(str));

sc.close();

}

}

1. **Tail recursion to calculate sum of array elements.**

Ans: public class Program9 {

private static int arrSum(int[] arr, int n, int sum) {

if(n==0)

return sum;

return arrSum(arr, n-1, sum+ arr[n-1]);

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter no of elements: ");

int n = sc.nextInt();

int arr[] = new int[n];

for(int i=0; i<n; i++) {

arr[i]= sc.nextInt();

}

System.out.println(arrSum(arr, n, 0));

}

}

1. **Recursive function to check if a string is palindrome.**

Ans: public class Program10 {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the string: ");

String str = sc.nextLine();

if(palindrome(str))

System.out.println("Yes");

else

System.out.println("No");

sc.close();

}

private static boolean palindrome(String str) {

int n = str.length();

if(n==0)

return true;

return palin(str, 0, n-1);

}

private static boolean palin(String str, int i, int j) {

if(i==j)

return true;

if(str.charAt(i) != str.charAt(j))

return false;

if(i<j+1)

return palin(str, i+1, j-1);

return true;

}

}

1. **Print Fibonacci Series in reverse order using Recursion.**

Ans: public class Program11 {

private static void reverse(int n) {

int a[]= new int[n];

a[0]=0;

a[1]=1;

for(int i=2; i<n; i++) {

a[i] = a[i-2]+a[i-1];

}

for(int i=n-1; i>=0; i--) {

System.out.print(a[i]+" ");

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the number: ");

int n = sc.nextInt();

reverse(n);

}

}