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

**public class PrintNumbers {**

**// Function to print numbers from 1 to n**

**public static void printNumbers(int n) {**

**if (n > 0) {**

**printNumbers(n - 1); // Recursively call with n-1**

**System.out.println(n); // Print the current number**

**}**

**}**

**public static void main(String[] args) {**

**int n = 10; // Change n to any positive integer**

**printNumbers(n);**

**}**

**}**

==============================================================================

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

**public class Main {**

**public static int sumOfNaturalNumbers(int n) {**

**if (n == 0)**

**return 0;**

**else**

**return n + sumOfNaturalNumbers(n - 1);**

**}**

**public static void main(String[] args) {**

**int n = 10;**

**int sum = sumOfNaturalNumbers(n);**

**System.out.println("Sum of natural numbers: " + sum);**

**}**

**}**

==============================================================================

1. **Mean of Array using Recursion**

**public class MeanOfArray {**

**// Recursive function to calculate sum of array elements**

**public static int arraySum(int[] arr, int n) {**

**if (n <= 0) {**

**return 0;**

**}**

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

**}**

**// Recursive function to calculate mean of array elements**

**public static double arrayMean(int[] arr, int n) {**

**if (n <= 0) {**

**return 0;**

**}**

**return (double) arraySum(arr, n) / n;**

**}**

**public static void main(String[] args) {**

**int[] array = {1, 2, 3, 4, 5};**

**int length = array.length;**

**double mean = arrayMean(array, length);**

**System.out.println("Mean of the array: " + mean);**

**}**

**}**

**===========================================================================**

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

**public class ArraySum {**

**// Recursive function to calculate the sum of array elements**

**public static int arraySum(int[] arr, int index) {**

**// Base case: if index reaches the end of the array**

**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 array elements: " + sum);**

**}**

**}**

========================================================================

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

**public class DecimalToBinary {**

**// Method to convert decimal to binary using recursion**

**public static String decimalToBinary(int decimal) {**

**// Base case: if the decimal number is 0, return "0"**

**if (decimal == 0) {**

**return "0";**

**}**

**return decimalToBinary(decimal / 2) + decimal % 2;**

**}**

**public static void main(String[] args) {**

**int decimalNumber = 25; // Example decimal number**

**String binary = decimalToBinary(decimalNumber);**

**System.out.println("Binary representation of " + decimalNumber + " is: " + binary);**

**}**

**}**

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

**import java.util.\*;**

**class HelloWorld {**

**public static int getSum(int n)**

**{**

**if (n == 0)**

**return 0;**

**return (n % 10 + getBinary(n / 10));**

**}**

**public static void main(String[] args) {**

**Scanner sc = new Scanner(System.in);**

**System.out.print("Enter you number binary number = ");**

**int num = sc.nextInt();**

**int x=getSum(num);**

**System.out.print("Digit Sum = "+x);**

**}**

**}**

==============================================================================

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

**public class ReverseString {**

**public static void main(String[] args) {**

**String str = "Hello World!";**

**System.out.println("Original String: " + str);**

**System.out.print("Reversed String: ");**

**reverse(str);**

**}**

**public static void reverse(String str) {**

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

**System.out.print(str);**

**} else {**

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

**reverse(str.substring(0, str.length() - 1));**

**}**

**}**

**}**

**==============================================================================**

1. **Program for length of a string using recursion**

**public class StringLength {**

**public static void main(String[] args) {**

**String str = "Hello World!";**

**int length = findLength(str);**

**System.out.println("Length of the string: " + length);**

**}**

**public static int findLength(String str) {**

**if (str == null || str.isEmpty()) {**

**return 0;**

**} else {**

**return 1 + findLength(str.substring(1));**

**}**

**}**

**}**

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

**public class ArraySum {**

**public static void main(String[] args) {**

**int[] arr = {1, 2, 3, 4, 5};**

**int sum = sumArray(arr, 0, 0);**

**System.out.println("Sum of array elements: " + sum);**

**}**

**public static int sumArray(int[] arr, int index, int sumSoFar) {**

**if (index == arr.length) {**

**return sumSoFar;**

**} else {**

**return sumArray(arr, index + 1, sumSoFar + arr[index]);**

**}**

**}**

**}**

**==============================================================================**

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

**public class Palindrome {**

**public static void main(String[] args) {**

**String str1 = "radar";**

**String str2 = "hello";**

**System.out.println(str1 + " is a palindrome: " + isPalindrome(str1));**

**System.out.println(str2 + " is a palindrome: " + isPalindrome(str2));**

**}**

**public static boolean isPalindrome(String str) {**

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

**return true;**

**} else if (str.charAt(0) == str.charAt(str.length() - 1)) {**

**return isPalindrome(str.substring(1, str.length() - 1));**

**} else {**

**return false;**

**}**

**}**

}

=========================================================================

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

**public class ReverseFibonacci {**

**public static void main(String[] args) {**

**int n = 10; // Number of Fibonacci numbers to print in reverse order**

**System.out.println("Fibonacci series in reverse order:");**

**reverseFibonacci(n);**

**}**

**public static void reverseFibonacci(int n) {**

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

**System.out.print(fibonacci(i) + " ");**

**}**

**}**

**public static int fibonacci(int n) {**

**if (n <= 1) {**

**return n;**

**} else {**

**return fibonacci(n - 1) + fibonacci(n - 2);**

**}**

**}**

**}**