**Question 1:Print 1 to n without using loops**

**--------------------------------------------------------------**

**--------------------------------------------------------------**

**public class PrintNumbers {**

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

**if (n > 0) {**

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

**System.out.print(n + " "); // Print current number**

**}**

**}**

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

**int n = 5; // Example: Print numbers from 1 to 5**

**printNumbers(n);**

**}**

**}**

**-------------------------------------------------------------**

**-------------------------------------------------------------**

**Question 2:Sum of natural numbers using recursion**

**public class SumOfNaturalNumbers {**

**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 = 5; // Example: Sum of first 5 natural numbers**

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

**}**

**}**

**--------------------------------------------------------------**

**--------------------------------------------------------------**

**Question 3:Mean of Array using Recursion**

**public class MeanOfArray {**

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

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

**return 0;**

**} else {**

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

**}**

**}**

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

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

**int n = arr.length; // Length of array**

**System.out.println("Mean of array: " + mean(arr, n));**

**}**

**}**

**--------------------------------------------------------------**

**--------------------------------------------------------------**

**Question 4:Sum of array elements using recursion**

**public class SumOfArrayElements {**

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

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

**return 0;**

**} else {**

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

**}**

**}**

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

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

**int n = arr.length; // Length of array**

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

**}**

**}**

**--------------------------------------------------------------**

**--------------------------------------------------------------**

**Question 5:Decimal to binary number using recursion**

**public class DecimalToBinary {**

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

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

**return "0";**

**} else if (n == 1) {**

**return "1";**

**} else {**

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

**}**

**}**

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

**int decimal = 10; // Example decimal number**

**System.out.println("Binary representation of " + decimal + ": " + decimalToBinary(decimal));**

**}**

**}**

**--------------------------------------------------------------**

**--------------------------------------------------------------**

**Question 6:. Sum of digit of a number using recursion**

**public class SumOfDigits {**

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

**if (n < 10) {**

**return n;**

**} else {**

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

**}**

**}**

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

**int num = 12345; // Example number**

**System.out.println("Sum of digits of " + num + ": " + sumOfDigits(num));**

**}**

**}**

**--------------------------------------------------------------**

**--------------------------------------------------------------**

**Question 7:Print reverse of a string using recursion**

**public class ReverseString {**

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

**if (str.isEmpty()) {**

**return str;**

**} else {**

**return reverse(str.substring(1)) + str.charAt(0);**

**}**

**}**

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

**String str = "hello"; // Example string**

**System.out.println("Reverse of \"" + str + "\": " + reverse(str));**

**}**

**}**

**--------------------------------------------------------------**

**--------------------------------------------------------------**

**Question 8:Program for length of a string using recursion**

**public class StringLength {**

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

**if (str.isEmpty()) {**

**return 0;**

**} else {**

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

**}**

**}**

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

**String str = "hello"; // Example string**

**System.out.println("Length of \"" + str + "\": " + length(str));**

**}**

**}**

**--------------------------------------------------------------**

**--------------------------------------------------------------**

**Question 9:Tail recursion to calculate sum of array elements**

**public class TailRecursionSum {**

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

**if (n < 0) {**

**return accumulator;**

**} else {**

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

**}**

**}**

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

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

**int n = arr.length; // Length of array**

**System.out.println("Sum of array elements: " + sum(arr, n - 1, 0));**

**}**

**}**

**--------------------------------------------------------------**

**--------------------------------------------------------------**

**Question 10:Recursive function to check if a string is palindrome**

**public class PalindromeCheck {**

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

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

**return true;**

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

**return false;**

**} else {**

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

**}**

**}**

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

**String str = "madam"; // Example string**

**System.out.println("Is \"" + str + "\" palindrome? " + isPalindrome(str));**

**}**

**}**

**--------------------------------------------------------------**

**--------------------------------------------------------------**

**Question 11:Print Fibonacci Series in reverse order using Recursion**

**public class FibonacciReverse {**

**public static void printFibonacciReverse(int n, int a, int b) {**

**if (n > 0) {**

**printFibonacciReverse(n - 1, b, a + b);**

**System.out.print(a + " ");**

**}**

**}**

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

**int n = 5; // Example: Print first 5 Fibonacci numbers in reverse order**

**int a = 0, b = 1; // Initial Fibonacci numbers**

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

**printFibonacciReverse(n, a, b);**

**}**

**}**

**--------------------------------------------------------------**

**--------------------------------------------------------------**