1. Count the number of digits in the given number.

import java.util.Scanner;  
  
public class CountNumberOfDigit {  
 static int *count* = 0;  
 public static int countDigit(int n){  
  
 if(n>0){  
 *count*++;  
 *countDigit*(n/10);  
 }  
 return *count*;  
 }  
  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.print(**"Enter the number : "**);  
 int n = sc.nextInt();  
 System.*out*.print(**"Number of terms in the digit are : "**);  
 int ans = *countDigit*(n);  
 System.*out*.println(ans);

}  
}

1. Find the power of 2 using recursion. Also take the input from the user.

public class PowerOf2 {  
 static int *count* = 0;  
 public static int powerOf2(int n){  
 if(n > 0){  
 if((n & 1) != 1){  
 *count*++;  
 }  
 *powerOf2*(n>>1);  
 }  
 return *count*;  
 }  
  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 int n = sc.nextInt();  
 System.*out*.println(*powerOf2*(n));  
 }  
}

1. Fibonacci series using recursion

import java.util.Scanner;  
  
public class FabonacciSeries {  
 static int *n1*=0,*n2*=1,*n3*=0;  
 static void printFibonacci(int count){  
 if(count>0){  
 *n3* = *n1* + *n2*;  
 *n1* = *n2*;  
 *n2* = *n3*;  
 System.*out*.print(**" "**+*n3*);  
 *printFibonacci*(count-1);  
 }  
 }  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 int n = sc.nextInt();  
 System.*out*.print(*n1* + **" "**+*n2*);  
 *printFibonacci*(n-2);  
 }  
}

1. Sum of n numbers using recursion

import java.util.Scanner;  
  
public class SumUsingRecursion {  
 static int *sum* = 0;  
 public static int sumFun(int n){  
// if(n == 0){  
// return 0;  
// }  
 if(n>0){  
 *sum* = n + *sumFun*(n-1);  
 }  
 return *sum*;  
 }  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 int n = sc.nextInt();  
 System.*out*.println(*sumFun*(n));  
 }  
}