1. Print 1 to n without using loops

import java**.**util**.**Scanner;

**public** **class** Ass2\_1 {

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

        Scanner sc = new Scanner(System.in);

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

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

        printNumber(1,num);

    }

**private** **static** **void** printNumber(**int** start, **int** end) {

        if (start > end) {

            return;

        }

        System.out.println(start);

        printNumber(start + 1, end);

    }

}

1. Sum of natural numbers using recursion

import java**.**util**.**Scanner;

**public** **class** Ass2\_2 {

**private** **static** **int** sumOfNaturalNumbers(**int** start, **int** end) {

        if (start ==  end) {

            return end;

        }

        return start+ sumOfNaturalNumbers(start + 1, end);

    }

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

         Scanner sc = new Scanner(System.in);

**int** num = 5;

        System.out.println(sumOfNaturalNumbers(1,num));

    }

}

1. Mean of Array using Recursion

**public** **class** Ass2\_3

{

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

    {

**int**[] arr = {3,6,2,4,1};

        System.out.println(findAverage(arr,0)/arr.length);

    }

**private** **static** **int** findAverage(**int**[] arr, **int** i) {

        if(i==arr.length-1)

        {

            return arr[i];

        }

        return arr[i]+findAverage(arr, i+1);

    }

}

1. Sum of array elements using recursion

**public** **class** Ass2\_4 {

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

**int**[] arr = {3,6,2,4,1};

        System.out.println(sumOfArray(arr,0));

    }

**private** **static** **int** sumOfArray(**int**[] arr, **int** i) {

        if(i==arr.length-1)

        {

            return arr[i];

        }

        return arr[i]+sumOfArray(arr, i+1);

    }

}

1. Decimal to binary number using recursion

**public** **class** Ass2\_9 {

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

**int** num = 11 ;

        String conversion = decimalToBinary(num);

        System.out.println(conversion);

    }

**private** **static** String decimalToBinary(**int** decimal)

    {

        if (decimal == 0) {

            return "";

        }

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

    }

}

1. Sum of digit of a number using recursion
2. **public** **class** Ass2\_11 {
3. **public** **static** **void** main(String[] args) {
5. **int** num = 1234;
6. System.out.println(sumOfDigits(num));
7. }
8. **private** **static** **int** sumOfDigits(**int** num) {
9. if(num == 0)
10. return num % 10;
11. return ( num % 10 ) + sumOfDigits(num / 10);
12. }
13. }
14. Print reverse of a string using recursion

**public** **class** Ass2\_5 {

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

        String a = "abcd";

        System.out.println("Reverse string is: " + reverseString(a, a.length()));

    }

**private** **static** String reverseString(String a, **int** length) {

        if(a.length() == 0)

            return "";

*//System.out.println(a.charAt(length-1));*

        return a.charAt(length-1) + reverseString(a.substring(0, length-1), length-1);

    }

}

1. Program for length of a string using recursion
2. **public** **class** Ass2\_6 {
3. **public** **static** **void** main(String[] args) {
4. String a = "abcd";
5. System.out.println(findLengthOfString(a));
6. }
7. **private** **static** **int** findLengthOfString(String a)
8. {
9. if(a.equals(""))
10. return 0;
11. return 1 + findLengthOfString(a.substring(1));
12. }
13. }

8. Tail recursion to calculate sum of array elements.

**public** **class** Ass2\_7 {

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

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

**int** sum = calculateSum(array, array.length-1, 0);

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

    }

**private** **static** **int** calculateSum(**int**[] array, **int** index, **int** partialSum) {

        if(index == 0)

            return partialSum;

*//System.out.println(partialSum + array[index]);*

        return  calculateSum(array, index - 1, partialSum + array[index]);

    }

}

9. Recursive function to check if a string is palindrome

**public** **class** Ass2\_8 {

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

        String a = "aba";

        System.out.println("Reverse string is: " + isPalindrome(a) );

    }

**private** **static** **boolean** isPalindrome(String str)

    {

        String target = str;

        return target.equals(reverseString(str, str.length())) ;

    }

**private** **static** String reverseString(String a, **int** length) {

        if(a.length() == 0)

            return "";

*//System.out.println(a.charAt(length-1));*

        return a.charAt(length-1) + reverseString(a.substring(0, length-1), length-1);

    }

}

10. Print Fibonacci Series in reverse order using Recursion

**public** **class** Ass2\_10 {

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

**int** num = 10;

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

        {

            System.out.println(fibo(i));

        }

    }

**private** **static** **int** fibo(**int** num) {

       if(num == 0 || num == 1)

            return num;

        return fibo( num - 1) + fibo(num - 2);

    }

}