Q.1. Print 1 to n without using loops

//WAP to print 1 to n natural numbers without using loops:

import java.util.\*;

public class A2Q1 {

static int baseNum = 0;

static void printLoop(int num)

{

if(baseNum < num)

{

baseNum += 1;

System.out.format("%d ", baseNum);

printLoop(num);

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

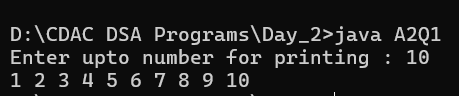
System.out.format("Enter upto number for printing : ");

int n = sc.nextInt();

printLoop(n);

}

}



Q.2. Sum of natural numbers using recursion

//WAP to add 1 to n natural numbers without using loops:

import java.util.\*;

public class A2Q2 {

static int baseNum = 0;

static int sum = 0;

static int sumLoop(int num)

{

if(baseNum < num)

{

baseNum += 1;

sum = sum + baseNum;

sumLoop(num);

}

return sum;

}

public static void main(String[] args) {

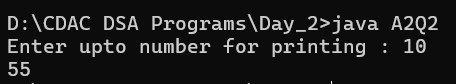
Scanner sc = new Scanner(System.in);

System.out.format("Enter upto number for printing : ");

int n = sc.nextInt();

int summation = sumLoop(n);

System.out.format("%d ", summation); } }



Q.3. Mean of Array using Recursion & 3. Sum of array elements using recursion

//WAP to add array elements without using loops:

import java.util.\*;

public class A2Q3 {

static int sum = 0;

static int sumLoop(int arr[], int N) { if (N <= 0) { return 0; }

return sumLoop(arr, N - 1) + arr[N - 1]; }

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.format("Enter array size : ");

int arrSize = 0;

arrSize = sc.nextInt();

int arr[] = new int[arrSize];

System.out.format("Enter array elemnts : ");

for(int i = 0; i < arrSize; i++)

{ arr[i] = sc.nextInt(); }

int N = 0;

N = arr.length;

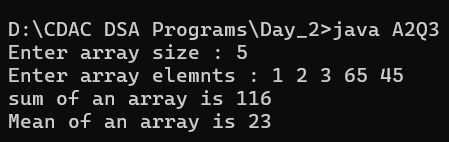
int summation = sumLoop(arr, N);

System.out.format("sum of an array is %d\n", summation);

int mean = 0;

mean = summation / arrSize;

System.out.format("Mean of an array is %d\n", mean); } }



Q.4. Decimal to binary number using recursion

//WAP to convert decimal number into binary :

import java.util.\*;

public class A2Q4 {

static int find(int decimal\_number)

{

if (decimal\_number == 0)

return 0;

else

return (decimal\_number % 2 + 10 \*

find(decimal\_number / 2));

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

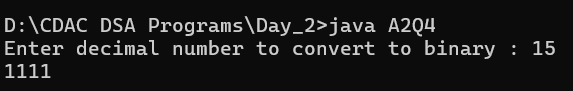
System.out.print("Enter decimal number to convert to binary : ");

int decimal\_number = sc.nextInt();

System.out.println(find(decimal\_number));

}

}



Q.5. Sum of digit of a number using recursion

//WAP to find Sum of digit of a number using recursion :

import java.util.\*;

public class A2Q5 {

static int sum\_of\_digit(int n)

{

if (n == 0)

return 0;

return (n % 10 + sum\_of\_digit(n / 10));

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter number to find its digits sum : ");

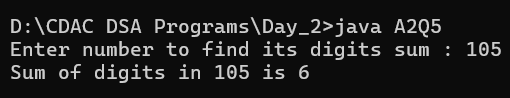
int num = sc.nextInt();

int result = sum\_of\_digit(num);

System.out.println("Sum of digits in " + num + " is " + result);

}

}



Q.6. Print reverse of a string using recursion

//WAP to Print reverse of a string using recursion :

import java.util.\*;

public class A2Q6 {

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) {

Scanner sc = new Scanner(System.in);

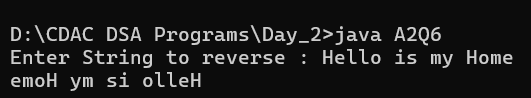
System.out.print("Enter String to reverse : ");

String str = sc.nextLine();

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

}

}



Q.7. Program for length of a string using recursion

//WAP for length of a string using recursion :

import java.util.\*;

public class A2Q7 {

private static int recLen(String str)

{

if (str.equals(""))

return 0;

else

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

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

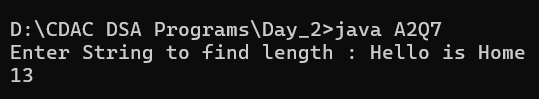
System.out.print("Enter String to find length : ");

String str = sc.nextLine();

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

}

}



Q.8. Tail recursion to calculate sum of array elements.

//WAP of Tail recursion to calculate sum of array elements:

import java.util.\*;

public class A2Q8 {

static int arrSum(int []array, int size, int sum)

{

if (size == 0)

return sum;

return arrSum(array, size - 1, sum + array[size - 1]);

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.format("Enter array size : ");

int arrSize = 0;

arrSize = sc.nextInt();

int arr[] = new int[arrSize];

System.out.format("Enter array elemnts : ");

for(int i = 0; i < arrSize; i++)

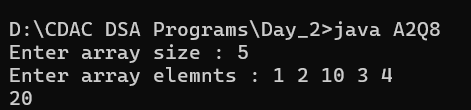
{

arr[i] = sc.nextInt();

}

int size = arr.length;

System.out.print(arrSum(arr, size, 0)); } }



Q.9. Recursive function to check if a string is palindrome

//WAP to check given String is palindrome or not :

import java.util.\*;

public class A2Q9 {

public static boolean isPalindrome(String s, int i){

if(i > s.length()/2)

return true ;

return s.charAt(i) == s.charAt(s.length()-i-1) && isPalindrome(s, i+1);

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter String to check if it is palindrome or not : ");

String str = sc.nextLine();

if (isPalindrome(str, 0))

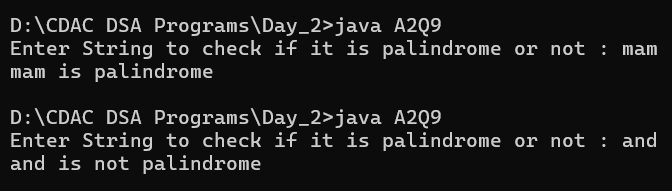
{ System.out.println(str + " is palindrome"); }

else

{ System.out.println(str + " is not palindrome"); }

}

}



Q.10. Print Fibonacci Series in reverse order using Recursion

//WAP to Print Fibonacci Series in reverse order using Recursion :

import java.util.\*;

public class A2Q10 {

static void rev\_fibo(int n, int a, int b){

if (n > 0) {

rev\_fibo(n - 1, b, a + b);

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

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter number to print Fibonacci Series in reverse : ");

int num = sc.nextInt();

rev\_fibo(num, 0, 1);

}

}

