Recursion Lab Exercises

# Instructions

* All lab exercises will be compiled as a lab report.
* All code should be well explained with comments.
* For recursion problems, when writing Java programs there should be the main method and a recursive method.

# Problem 1

Write a Java program based on recursion that will count down numbers from number n to 0 while printing the numbers out.

1. Write some kind of pseudocode/algorithm or really just the outline of the method.

function countdown(n)

print n  
  if n is zero

return  
  otherwise if n is greater than zero  
    call countdown with (n-1)

1. Write a complete Java program that will apply the method in (a)

/\*\* Coutdown function \*/

static void countdown(int n) {

    // Output the number

    System.out.println(n);

    // Base case

    if (n <= 0) {

        return;

    }

    // General case

    countdown(n-1);

}

# Problem 2

Write a program in Java to print the first 50 (1-50) numbers using recursion.

// Countup function

static void countup(int start, int end) {

    // Base case

    if (start > end){

        return;

    }

    // Output the number

    System.out.println(start);

    // Increment the number in the general case

    countup(++start, end);

}

# Problem 3

Write a program in Java to calculate the sum of numbers from 1 to n using recursion

// Sum up numbers from 1 to n inclusive

    static long sum(long n) {

        // Base case

        if (n <= 1) {

            return n;

        }

        // General case

        return n+sum(n-1);

    }

    // A faster implementation without recursion

    static int fastSum(int n) {

        double x = n;

        return (int)((x/2.0)\*(1+n));

    }

# Problem 4

Write a complete program in Java to print array elements using recursion.

/\*\*

 \* ArrayOutput

 \*/

public class ArrayOutput {

    /\* Basic array output using recursion

     \*/

    static <T> void \_outputArray (T arr[], int index) {

        // Base case

        if (index == arr.length) {

            return;

        }

        System.out.println(arr[index]);

        \_outputArray(arr, ++index);

    }

    /\* Array output function that hides recursive function\

     \* Works for any object type except primitives (int, double, long, char, ...)

     \*/

    static <T> void outputArray(T arr[]) {

        // Call the recursive function.

        \_outputArray(arr, 0);

    }

    public static void main (String args[]) {

        Integer arr[] = {1, 2, 3, 4, 5, 6};

        outputArray(arr);

        String arrS[] = {"Hello", "World!", "How are you?"};

        outputArray(arrS);

    }

}

# Problem 5

Write a program in Java to find the GCD of two numbers using recursion

public class GDC {

    static long getGDC(long num1, long num2) {

        int i = 2;

        boolean lcmFound = false;

        for (; i < 10; i++) {

            if (num1%i == 0 && num2 % i == 0) {

                lcmFound = true;

                break;

            }

        }

        if (!lcmFound) {

            return 1;

        }

        return i\*getGDC(num1/i, num2/i);

    }

    public static void main(String[] args) {

        System.out.println(getGDC(52, 144));

    }

}