PROGRAMACION II - GUIA DE EJERCICIOS DE RECURSIVIDAD

1.- Realizar un procedimiento que imprima de manera recursiva los números del 1 al 10.

2.- Realizar un procedimiento que imprima de manera recursiva la tabla del 9.

3.- Realizar un procedimiento que encuentre el número mayor de un vector de n números enteros de manera recursiva.

4.- The fibonacci sequence is a famous bit of mathematics, and it happens to have a recursive definition. The first two values in the sequence are 0 and 1 (essentially 2 base cases). Each subsequent value is the sum of the previous two values, so the whole sequence is: 0, 1, 1, 2, 3, 5, 8, 13, 21 and so on. Define a recursive fibonacci(n) method that returns the nth fibonacci number, with n=0 representing the start of the sequence.

fibonacci(0) → 0
fibonacci(1) → 1
fibonacci(2) → 1

5.- We have a number of bunnies and each bunny has two big floppy ears. We want to compute the total number of ears across all the bunnies recursively (without loops or multiplication).

bunnyEars(0) → 0
bunnyEars(1) → 2
bunnyEars(2) → 4

6.- We have bunnies standing in a line, numbered 1, 2, ... The odd bunnies (1, 3, ..) have the normal 2 ears. The even bunnies (2, 4, ..) we'll say have 3 ears, because they each have a raised foot. Recursively return the number of "ears" in the bunny line 1, 2, ... n (without loops or multiplication).

bunnyEars2(0) → 0
bunnyEars2(1) → 2
bunnyEars2(2) → 5

7.- Given a non-negative int n, return the count of the occurrences of 7 as a digit, so for example 717 yields 2. (no loops). Note that mod (%) by 10 yields the rightmost digit (126 % 10 is 6), while divide (/) by 10 removes the rightmost digit (126 / 10 is 12).

count7(717) → 2
count7(7) → 1
count7(123) → 0

8.- Given n of 1 or more, return the factorial of n, which is n * (n-1) * (n-2) ... 1. Compute the result recursively (without loops).

factorial(1) → 1
factorial(2) → 2

factorial(3) → 6

9.- Given a non-negative int n, return the sum of its digits recursively (no loops). Note that mod (%) by 10 yields the rightmost digit (126 % 10 is 6), while divide (/) by 10 removes the rightmost digit (126 / 10 is 12).

sumDigits(126) → 9
sumDigits(49) → 13
sumDigits(12) → 3

10.- Given base and n that are both 1 or more, compute recursively (no loops) the value of base to the n power, so powerN(3, 2) is 9 (3 squared).

powerN(3, 1) → 3
powerN(3, 2) → 9
powerN(3, 3) → 27