**Taller #5.**

**RECURSIÓN**

1. **Condiciones generales**

* use ***lowerCamelCase.***
* Para la entrega, suba al Bb en un archivo comprimido con su nombre completo, los archivos (.c, y.exe) que tienen la solución al taller.
* Cree las funciones y procedimientos que considere necesarios para desarrollas los diferentes puntos del taller. Crear toda la solución de nuestro programa directamente en el main() no es una buena práctica de programación. Desacoplar las funcionalidades de sus programas no sólo mejoran la estética, también mejora el mantenimiento del código.

1. **Puntos para desarrollar en el taller… Espero que sea entretenido ☺**
2. Para elegir qué opción del taller poner a correr, usted deber crear un Menú. Pero no es cualquier menú, su menú no puede ser hecho de manera iterativa, debe ser hecho con recursión. Valga decir que la recursión no es la forma más eficiente de hacer un menú, y lo sé, pero es para efectos prácticos y de aplicar el concepto.
3. Sume los múltiplos de 3entre un rango A y B decididos por el usuario. (indique los tipos de recursividad usados para la solución. Los puede poner como comentarios en el código).
4. Realice una función recursiva que permita calcular el MCD (máximo común divisor) de dos números enteros positivos utilizando el Algoritmo de Euclides:

Algoritmo de Euclides: Dados dos números enteros positivos m y n, tal que m > n, para encontrar su máximo común divisor (es decir, el mayor entero positivo que divide a ambos):

* Dividir m por n para obtener el resto r (0 ≤ r < n)
* Si r = 0, el MCD es n
* Si no, el máximo común divisor es MCD(n,r)

1. Con un algoritmo recursivo, encuentre el valor mayor de un arreglode enteros con 10 valores (cree su arreglo con lista inicializadora).
2. Realice la solución a la **serie Fibonacci** (indique los tipos de recursividad usados para la solución. Los puede poner como comentarios en el código) Recuerde que Fibonacci de 1 es 1, Fibonacci de 2 es 1 y para los demás términos Fibonacci es la suma de sus últimos 2 términos (n-1 + n-2)
3. Una Matriz identidad, es una matriz cuadrada que posee unos (1) en su diagonal principal y ceros en el resto de sus valores. Su tarea consiste en realizar una función recursiva que se encargue de evaluar una matriz, de manera que retorne 1 en caso de ser matriz identidad, y 0 en caso de no serlo. (su función no necesita validar si la matriz es cuadrada, partimos de la base de que ya existiera una validación anterior que nos asegura que siempre ingresa una matriz cuadrada)