



## Práctica 3 – Recursividad

1. Desarrollar funciones (algunas pueden ser void) para:
  - a) calcular el producto de dos números enteros positivos a partir de sumas sucesivas.
  - b) calcular cociente y resto entre dos números enteros positivos a partir de restas sucesivas
  - c) hallar el máximo elemento de un vector de enteros (realizar una versión void y otra int)
2. Recodificar en el lenguaje C los siguientes ejercicios de Programación I desarrollados en Pascal:
  - a) Sea V un arreglo de enteros, se pide:
    - I. Calcular e informar la suma de sus elementos.
    - II. Mostrar sus elementos.
    - III. Mostrar sus elementos en orden inverso.
  - b) Sea A un arreglo de reales y un valor X real ingresado por teclado, se pide determinar en que posición se encuentra la primera ocurrencia de X o -1 en caso de no encontrarlo. Implementar:
    - I. Considerando el arreglo desordenado, una búsqueda lineal recursiva.
    - II. Para un arreglo ordenado en forma ascendente, una búsqueda binaria recursiva.
  - c) Crear una función recursiva que busque el elemento mínimo de una matriz cuadrada.
  - d) Desarrollar un procedimiento recursivo que devuelva en un arreglo los máximos de cada fila de una matriz de MxN.
  - e) Dado dos arreglos de enteros A y B ordenados ascendentemente, mezcle ambos arreglos en otro C manteniendo el orden pero sin pasar los elementos repetidos.
3. Dado un vector A de N elementos, obtener el promedio de sus elementos.
4. Dada una matriz A de NxM float, retornar la cantidad de veces que un valor X aparece en A (realizar una versión void y otra int)
5. Determinar mediante una función “lógica” si una matriz cuadrada de dimensión NxN es simétrica ( $a_{ij} = a_{ji}$ ) con respecto a su diagonal.
6. Dada una matriz de NxM enteros, determinar si la suma de los elementos de cada fila es mayor a la suma de los elementos de la fila anterior. Para la primera fila se debe cumplir que la suma de sus elementos sea par.