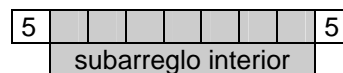


**Práctico 5: Recursión**

**Referencias**

- Material de las clases Teóricas de Introducción a la Programación II.
- Bishop, Judy; Pascal Precisely, 3rd Edition, Addison-Wesley, 1993. Capítulo 3.
- Shackelford, Russell L.; Introduction to Computing and Algorithms, Addison-Wesley, 1998. Capítulos 4 y 6.
- Gries, D.; A Science of Programming, 1998. Capítulo 18.

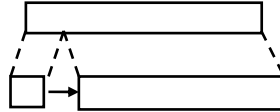
- 1) Una función es recursiva si su código contiene invocaciones a sí misma.  
Teniendo en cuenta que  $\text{factorial}(N) = N * \text{factorial}(N-1)$  para  $N > 0$  y  $\text{factorial}(0) = 1$ , escriba el código de la función que dado un número retorna su factorial.
- 2) Calcular recursivamente el Máximo Común Divisor de dos enteros no negativos basándose en las siguientes fórmulas matemáticas.
  - a.-  $\text{MCD}(X,X) = X$   
 $X < Y \Rightarrow \text{MCD}(X,Y) = \text{MCD}(Y,X)$   
 $X > Y \Rightarrow \text{MCD}(X,Y) = \text{MCD}(X-Y,Y)$
  - b.-  $\text{MCD}(X,0) = X$   
 $X < Y \Rightarrow \text{MCD}(X,Y) = \text{MCD}(Y,X)$   
 $X \geq Y \Rightarrow \text{MCD}(X,Y) = \text{MCD}(Y,\text{MOD}(X,Y))$Compare las soluciones de este ejercicio y el anterior con la versión iterativa realizada en el ejercicio correspondiente del Práctico 6 de Introducción a la Programación I.
- 3) Codifique una función que calcule el factorial del MCD de dos números dados, según el algoritmo 2.b. Hacer la secuencia de llamados y retornos de procedimientos con valores 28 y 20.
- 4) Para verificar que un arreglo es capicúa de manera recursiva se sigue la siguiente estrategia. Un arreglo es capicúa si el primer elemento es igual al último y si el “subarreglo” que queda entre estos dos elementos también es capicúa. Ejemplo:



Para poder plantearlo recursivamente, la clave está en poder aplicar la misma estrategia para verificar si el subarreglo interior es capicúa. Piense cómo solucionar este último problema, plantee cuándo “cortar” la recursión y codifique la solución.

- 5) Verificar recursivamente que una matriz cuadrada de 33 caracteres de lado es palíndroma (capicúa en todas sus filas y columnas). Versión recursiva del ejercicio correspondiente del Práctico 8 de Introd. a la Prog. I.
- 6) Plantee y codifique una estrategia recursiva para grabar los datos de un archivo en otro de tal forma que los mismos queden en orden inverso.
- 7) Piense los casos de verificación que podría incorporar para probar el ejercicio anterior incluyendo los resultados esperados.
- 8) Incorporó los siguientes casos en el ejercicio anterior? A) archivo vacío. B) archivo con sólo dos datos iguales. Pase el programa del ejercicio 6 a la computadora, y compare los resultados esperados con los obtenidos en la ejecución y los resultantes de una “ejecución manual”.

- 9) Una forma interesante de conceptualizar una lista para pensar soluciones recursivas es pensarla como “cabeza-cola” donde la cabeza es el primer elemento de la lista y la cola el resto. Ej:



La ventaja es que la cola es una lista y por lo tanto podemos pensar recursivamente, es decir aplicar la misma estrategia para la cola que se aplica a toda la lista.

Ejemplo: sumar los elementos de una lista vinculada.

Solución: sumo el valor de la cabeza de la lista con el resultado de la función aplicada al resto (la cola) de la lista. Codifique la solución.

- 10) Piense recursivamente y codifique una solución para devolver el menor de los elementos de una lista vinculada.

- 11) *Merge sort*: este método de ordenamiento trabaja de la siguiente manera: dada una lista la divido en dos mitades, ordeno cada mitad (para lo cual puedo aplicar recursión) y después intercalo las mitades. (el procedimiento para intercalar es muy simple comparo las cabezas de las listas y tomo la menor). Ej:

6	2	5	8	3	1	7	9	4
---	---	---	---	---	---	---	---	---

*separo*

6	2	5	8
---	---	---	---

3	1	7	9	4
---	---	---	---	---

*ordeno cada mitad*

2	5	6	8
---	---	---	---

1	3	4	7	9
---	---	---	---	---

*intercalo*

1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---

Implemente la solución recursivamente.