

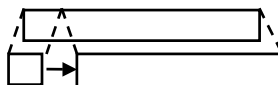
Práctico 5: Recursión

- 1) Se dice que una función es recursiva si su código genera invocaciones a si misma. Teniendo en cuenta que $\text{fact}(4) = 4 * \text{fact}(3)$, y que, en general $\text{fact}(N) = N * \text{fact}(N-1)$, $N > 0$. 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.- $\text{MCD}(X,X) = X$
 $\text{MCD}(X,Y) = \text{MCD}(Y,X)$
 $X > Y \Rightarrow \text{MCD}(X,Y) = \text{MCD}(X-Y,Y)$
 - b.- $\text{MCD}(X,0) = X$
 $\text{MCD}(X,Y) = \text{MCD}(Y,X)$
 $\text{MCD}(X,Y) = \text{MCD}(Y, \text{MOD}(X,Y))$Compare las soluciones de este ejercicio y el anterior con la versión iterativa realizada en los ejercicios 12 y 13 del Práctico 6 de Introd. a la Prog. I.
- 4) 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.
- 5) 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. Ej:



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

- 6) Verificar recursivamente que una matriz cuadrada de 33 caracteres de lado es palíndroma (capicúa en todas sus filas y columnas). Versión recursiva del ejercicios 10 Práctico 8 de Introd. a la Prog. I.
- 7) Para invertir el orden de los elementos de una archivo (ej 6 del Práctico 2) recursivamente se plantea la siguiente estrategia: “leo un elemento en la variable A, llamo a un procedimiento que invierta y grabe los elementos que quedan en el arreglo y luego escribo la variable A”. Claramente, el elemento que leí primero lo escribo último, por lo que si confié en que el procedimiento que invierte el resto funciona, entonces la solución final también funciona. Obviamente, el problema de invertir el resto lo resuelvo de la misma manera (leo el primero, invierto lo que queda y grabo después). Codifique la solución.
- 8) 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 una solución A como procesar la cabeza y aplicar la cola la misma solución A.

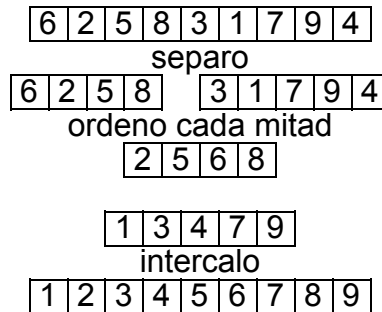
Ejemplo: sumar los elementos de una lista vinculada.

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

Codifique la solución.

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



Implemente la solución recursivamente.