



## Listas y recursión

Resuelva los siguientes ejercicios y entreguelos todos en un solo archivo de texto plano, con extensión *.prolog* y coloque como comentario inicial de cada problema, el enunciado del mismo. Puede usar más de un predicado para cada problema. No olvide comentar adecuadamente...

Construya predicados *Prolog*, etiquetados como se indica con color azul en cada problema y que resuelvan las tareas solicitadas. Respete las restricciones marcadas entre corchetes y coteje contra los ejemplos proporcionados en cada problema.

- 1) [] **contiene\_numero/1**. Verificar si una lista contiene algún elemento numérico.

**contiene\_numero(<lista>).**

Verdadero si **<lista>** es una lista que contiene algún elemento numérico,  
Falso en cualquier otro caso.

Ejemplos: *contiene\_numero([a,b,c,d]). → false.*  
*contiene\_numero(23). → false.*  
*contiene\_numero([a,b,3,d,e]). → true.*

- 2) [] **inserta\_ceros/2**. Intercalar ceros después de cada elemento original.

**inserta\_ceros(<lista>, <respuesta>).**

Verdadero si **<respuesta>** es una lista con los mismos elementos que **<lista>**, pero con un cero agregado después de cada elemento original. La lista vacía debe conservarse.

Ejemplos: *inserta\_ceros([a,2,c,3], R). R = [a,0,2,0,c,0,3,0].*  
*inserta\_ceros([], R). R = [].*

- 3) [sin usar **append**] **rota/3**. Rotar los elementos de una lista algún número de posiciones hacia la derecha.

**rota(<lista>, <n>, <respuesta>).**

Verdadero si **<respuesta>** es una lista con los mismos elementos que **<lista>**, pero rotados hacia la derecha **<n>** posiciones.

Ejemplos: *rota([a,b,c,d], 1, R). R = [d,a,b,c].*  
*rota([a,b,c,d], 2, R). R = [c,d,a,b].*  
*rota([a,b,c,d], 3, R). R = [b,c,d,a].*  
*rota([a,b,c,d], 4, R). R = [a,b,c,d].*

- 4) [sin usar *reverse*, ni *append*] *reversa\_simple/2*. Invertir una lista.

*reversa\_simple(<lista>, <respuesta>).*

Verdadero si *<respuesta>* es la inversión de primer nivel de *<lista>*.

Ejemplos: *reversa\_simple([a,b,c,d], R)*. *R = [d,c,b,a]*.  
*reversa\_simple([[1,2],3,[a,b],c], R)*. *R = [c,[a,b],3,[1,2]]*.

- 5) [sin usar *select*] *inserta\_en/4*. Insertar un término arbitrario en alguna posición específica de una lista.

*inserta\_en(<término>, <lista>, <posición>, <resultado>).*

Verdadero si *<resultado>* es una lista con los mismos elementos que *<lista>* pero con *<término>* insertado en la posición *<posición>*, considerando el inicio de la lista como la posición 0.

Ejemplos: *inserta\_en(x, [a,b,c,d], 2, R)*. *R = [a,b,x,c,d]*.  
*inserta\_en(3, [a,b,c,d], 1, R)*. *R = [a,3,b,c,d]*.  
*inserta\_en(, [], 4, R)*. *false*.

- 6) [] *promedio\_parcial/3*. Calcular el promedio (media aritmética) de los primeros *n* elementos de una lista.

*promedio\_parcial(<lista>, <n>, <resultado>).*

Verdadero si *<resultado>* es un número que representa el promedio de los primeros *<n>* elementos de *<lista>*.

Ejemplos: *promedio\_parcial([2,4,6,8,10], 3, R)*. *R = 4*.  
*promedio\_parcial([a,b,c,d], 2, R)*. *false*.  
*promedio\_parcial([1,3,5,7], 4, R)*. *R = 4*.

- 7) [sin optimizar] *fibonacci/2*. Calcular cada término en la serie de Fibonacci.

*fibonacci(<n>, <resultado>).*

Verdadero si *<resultado>* es el número Fibonacci correspondiente a *<n>*.

Ejemplos: Los primeros elementos de la serie son:

$fib(0)=1.$   $fib(1)=1.$   $fib(2)=1.$   $fib(3)=2.$   $fib(4)=3.$   
 $fib(5)=5.$   $fib(6)=8.$   $fib(7)=13.$   $fib(8)=21.$   
 $fib(9)=34.$   $fib(10)=55.$

- 8) [] **simplifica/2.** Eliminar de una lista todos los elementos que se encuentren duplicados.

***simplifica(<lista>, <resultado>).***

Verdadero si **<resultado>** es una lista con los mismos elementos que **<lista>** pero con sólo una instancia de cada elemento.

Ejemplos: *simplifica([a,b,1,2,c,a,2,1,4], R).*  $R = [b,c,4].$   
*simplifica([1,1,2,2,3,3], R).*  $R = [].$

- 9) [] **depura/2.** Eliminar de una lista todos los elementos que NO se encuentren duplicados, cuando menos, una vez.

***depura(<lista>, <resultado>).***

Verdadero si **<resultado>** es una lista conteniendo sólo una instancia de cada elemento en **<lista>** que sí tenía repeticiones.

Ejemplos: *depura([a,b,1,2,c,a,2,1,4], R).*  $R = [a,1,2].$   
*depura([1,1,2,2,3,3], R).*  $R = [1,2,3].$   
*depura([1,2,3,4], R).*  $R = [].$

- 10) [] **maximo/2.** Identificar el mayor valor de entre aquellos contenidos en una lista.

***maximo(<lista>, <resultado>).***

Verdadero si **<resultado>** es el mayor valor numérico contenido en **<lista>**. No todos los elementos necesitan ser numéricos.

Ejemplos: *maximo([a,b,1,2,c,a,2,1,4], R).*  $R = 2.$   
*maximo([1,1,2,2,3,3], R).*  $R = 3.$   
*maximo([a,b,c,d], R).*  $R = 0.$