

Ejercicios de programación lógica pura

Ejercicio 1 Teniendo en cuenta la siguiente definición de número natural:

```
1 natural(0).  
2 natural(s(X)) :-  
3   natural(X).
```

Escribir un programa lógico puro con los siguientes predicados:

- a) **menor_que(X,Y)**, cierto sí y sólo si **X** es menor que **Y**.
- b) **par(X)**, cierto sí y sólo si **X** es par
- c) **suma(X,Y,Z)**, cierto sí y sólo si **Z** es la suma de **X** e **Y**.
- d) **producto(X,Y,Z)**, cierto sí y sólo si **Z** es el producto de **X** e **Y**.

Ejercicio 2 Utilizando los predicados del ejercicio anterior, escribir un programa lógico puro con los siguientes predicados:

- a) **suma_a_lista(L,N,SL)**, cierto sí y sólo si la lista **SL** es la lista que se obtiene al sumarle **N** a cada uno de los elementos de la lista **L**.
- b) **pares_lista(L,Ps)**, cierto sí y sólo si **Ps** son los números pares que se encuentran en la lista **L**.
- c) **acumula_lista(L,S)**, cierto sí y sólo si **S** es la suma de los elementos de la lista **L**.

Ejercicio 3 Diseña y escribe un programa lógico puro que, dado un índice (con la misma notación que el ejercicio anterior) y una lista devuelva el elemento de la lista que está en la posición indicada por el índice y la lista sin el elemento anterior. Por ejemplo, dado el índice **s(s(0))** y la lista **[a,b,c,d,e]** se debe devolver el elemento **b** y la lista **[a,c,d,e]**.

Ejercicio 4 Se pide al alumno que escriba un programa lógico puro para realizar el recorrido en anchura de un árbol binario, codificando los siguientes predicados, así como cualquier otro que sea necesario. El predicado principal **anchura(A,L)** es cierto si y solo si **L** es la lista de los nodos del árbol **A** en el orden en que se visitan en un recorrido en anchura (de izquierda a derecha).

El predicado auxiliar **hijos(As,Hs,Ns)** es cierto si y solo si **Hs** es la lista de los árboles hijos de la lista de árboles **As** y el par **Ns** es la lista de los nodos raíz de dichos árboles (**As**). Se debe utilizar la representación de listas habitual en Prolog y explicar la que se utilice para los árboles.

Ejercicio 5 Se pide al alumno que escriba un programa lógico puro para obtener la lista de nodos que estén a una profundidad dada. El predicado principal **nodos_profundidad(N,A,L)** es cierto si y solo si **L** es la lista de los nodos del árbol **A** que están a profundidad **N** el orden en que se visitan en un recorrido en anchura (de izquierda a derecha).

Sugerencia: reutilizar el predicado **hijos(As,Hs,Ns)** del ejercicio anterior.