Guía de Trabajos Prácticos IX – Programación Lógica II

1) Cree un programa en Prolog que permita calcular el factorial de un número con el predicado "factorial/2" en donde el primer argumento unifique con el número al que se le calculará el factorial, y el segundo con el factorial del mismo.

Se debe verificar que el número pasado como primer parámetro sea mayor a cero, en caso contrario, el predicado debe fallar.

Ej.: factorial(5, X). => X = 120.

2) Escribir un programa en Prolog "contar/3" que reciba como primer parámetro una lista de números. Se debe unificar el segundo argumento con la cantidad de elementos pares que encuentre (tomando el 0 como par) y el tercer argumento con la cantidad de elementos impares de la misma.

Si algún elemento de la lista no fuera un número entero el predicado debería fallar, para esto se puede utilizar el predicado predefinido "integer/1" que unifica solo si su argumento es un número entero.

Ejemplos:

contar([1, 2, 3], P, I).
P = 1
I = 2

contar([1, 2, a], P, I).
false

3) Escribir un predicado en Prolog que sume los elementos de una lista de números enteros recibidos como primer argumento y unifique el resultado de dicha suma con el segundo argumento.

Tener en cuenta

Ejemplos:

sumar([1, 2, 3], X).
X = 6

sumar([1, -2, 3], X).
X = 2

4) Escribir un programa en Prolog que aplane una lista. El predicado aplanar/2 recibe una lista cuyos elementos pueden ser otras listas.

No se deben utilizar predicados predefinidos si los hubiere.

Ejemplos:

aplanar([1, 3, 2], L).
L = [1, 2, 3].

aplanar([1, [3, 2]], L).
L = [1, 3, 2].

aplanar([1, [3, 2], [1, 5, 4], 3, [5, 1]], L).
L = [1, 3, 2, 1, 5, 4, 3, 5, 1].

5) Escribir un programa en Prolog que reciba dos listas de números, verifiquen que sean de la misma longitud, y luego retorne una lista con la suma elemento a elemento de ambas listas.

Ejemplos:

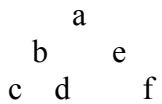
sumar([1, 2], [1], L). %falla porque son de distinto tamaño
false

sumar([1, 2], [5, 3], L).
L = [6, 5]

6) Escribir un programa en Prolog que recorra un árbol binario y determine la profundidad del mismo.

La representación del árbol será una lista con el siguiente formato: [I, N, D] en donde:
I es una lista que representa el subárbol de la rama izquierda
N es el valor del nodo raíz
D es una lista que representa el subárbol de la rama derecha

así el árbol:



estaría representado por [[[c], b, [d]], a, [[], e, [f]]]

prestar atención a que el las ramas vacías se representan con una lista vacía, y las hojas como un alista de un solo elemento.

7) El siguiente programa en Prolog calcula las permutaciones de los elementos de una lista.

1. Ejecute el mismo y escriba el resultado obtenido para per([1, 2, 3], L).
2. Explique en sus propios términos cual es la lógica que utiliza el programa para obtener las permutaciones.

```
ins(X, L, [ X | L ]).
ins(X, [ Y | L1 ], [ Y | L2 ]) :- ins(X, L1, L2).

per([],[]).
per([ X | L ], Lp) :- per(L, L1), ins(X, L1, Lp).
```