


Maestría en Ciencias de la Computación.
Programación Avanzada
Gustavo Márquez Flores.


Tercer Tarea.
Programación Lógica.

1.- Un árbol en *Prolog* se puede representar mediante el siguiente término :

`arb(I, X, D)`

done I es el subárbol izquierdo y D el subárbol derecho, los cuales se representan de la misma manera; X es el elemento raíz del árbol. Escribe los predicados necesarios para realizar las siguientes operaciones:

 `x_pert_arb(X, Arb)` : Cierta si X está en el árbol Arb

 `nods_arb(Arb, N)` : N es el número de nodos del árbol Arb

2.- Escribe un predicado que determine si tres cantidades determinan un triángulo y de qué tipo es:

`?-triangulo(9, 10, 9).`
Triángulo isósceles.

`?-triangulo(10, 10, 10).`
Triángulo equilátero.

`?-triangulo(10, 7, 15).`
Triángulo escaleno.

`?-triangulo(10, 4, 15).`
No es posible formar un triángulo.

3.- Escribe los siguientes predicados en *Prolog*:

*% Cuenta(Xs, N) : Cierta si N es el número de elementos distintos en
% la lista Xs .*

*% filtra(Xs, Ys) : Elimina de la lista Xs los elementos repetidos,
% obteniendo una nueva lista YS .*

4.- Escribe un programa para sumar matrices mediante una lista de listas:

`?- sumamatriz([[2,3],[4,5]], [[4,5],[6,7]], S).`

`S=[[6,8], [10,12]].`

5.- Escribe un predicado que obtenga el valor de la función de *Ackermann* para un valor específico:

`?-ackermann(1, 2, X).`
`X=4.`