

TAREA 5: PROLOG - Parte 2

Paradigmas de Programación

NOTA: Para todos los ejercicios se deben ejecutar metas (consultas) interesantes que demuestren la efectividad de la solución:

Fecha de Entrega: jueves 2 de octubre

1. Programe en PROLOG:

(a)

$$U_n = \begin{cases} 1 & \text{si } n = 0 \\ 0 & \text{si } n = 1 \\ -1 & \text{si } n = 2 \\ 5U_{n-1} - 3U_{n-2} - 9U_{n-3} & \text{si } n > 3 \end{cases}$$

(b) La función de Ackermann

$$A(N, M) = \begin{cases} 0 & \text{Si } M = 0 \\ 2M & \text{Si } N = 0 \\ 2 & \text{Si } M = 1 \\ A(N - 1, A(N, M - 1)) & \text{en otro caso} \end{cases}$$

(c) Una función que calcule recursivamente

$$f(x) = x^n$$

Debe leer de teclado X y N.

Debe imprimir el resultado.

2. Escriba un programa en PROLOG que contenga una lista de números reales y que efectúe lo siguiente:

- Imprime lista.
- Lee lista.
- Suma lista.
- Multiplica lista.
- Suma tres listas y el resultado lo deja en una cuarta lista.
- Producto de tres listas y el resultado lo deja en una cuarta lista.

- Ordena una lista
 - Eliminar(X,L1,L2) que elimina la primera aparición de X en la lista L1, el resultado lo deja en L2.
 - Subconjunto(L1,L2) "retorna" TRUE si la lista L1 está contenida en la lista L2.
 - Debe tener cláusulas de invocación.
3. Escriba un programa en PROLOG que para un objeto compuesto Empleado que tiene Cédula, Nombre, Horas_Trabajadas, Sueldo_por_hora, Sueldo = Horas_Trabajadas * Sueldo_por_Hora, permita:
- Leer sus datos.
 - Imprimir sus datos.
 - Almacenar los datos en una lista.
 - Imprimir la lista de empleados.
4. Realice un programa en PROLOG que calcule la series:
- $\sum_{i=a}^n (2i + 4)^2$
 - $\sum_{i=a}^n 2i^2 - i$
 - $\sum_{i=a}^n \frac{i+1}{2i^2+7i-3}$
5. Construya en PROLOG un predicado Menor(L,E) que relacione una lista con su menor elemento. Algunos ejemplos son:
- Menor([5 3 9 2 5 6], X) ----> X=2
 - Menor([4 1 7], 3) ----> False
6. Construya un predicado el PROLOG MenorIguar(L1,L2) que compare dos hileras representadas por listas. Algunos ejemplos son:
- MenorIguar([H, O, L, A], [H, O, Y]) -----> True
 - MenorIguar([M, E, S, A], [C, A, S, A]) -----> False
 - MenorIguar([E, L], [E, L, L, A]) -----> True