

Profesor: Dr. Oldemar Rodríguez Rojas

Paradigmas de Programación

Programación PROLOG - Parte 2

Fecha de Entrega: Jueves 14 de octubre - 8am

Instrucciones:

- La fecha de entrega es el día
- Cada día de atraso implicará una pérdida de 20 puntos.
- Las tareas son estrictamente de carácter individual, tareas iguales se les asignará cero puntos.
- Deben incluir los casos de prueba que se solicitan en los ejercicios.

TAREA NÚMERO 6

1. [10 puntos] Escriba una función recursiva en PROLOG para la función de Ackerman:

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

2. [10 puntos] Programe en PROLOG con recursión la siguiente función:

$$U_n = \begin{cases} 2U_{n-1} - 7U_{n-2} - 6U_{n-3} & \text{si } n \geq 3 \\ 1 & \text{si } n = 0 \\ 0 & \text{si } n = 1 \\ -1 & \text{si } n = 2 \end{cases}$$

3. [10 puntos] Programe en PROLOG funciones recursivas para las siguientes operaciones con listas:

- Imprime lista.
- Multiplica lista.
- Suma tres listas y el resultado lo deja en una cuarta lista.
- Subconjunto(L1, L2) retorna TRUE si la lista L está contenida en la lista L2.
- Eliminar(X, L1, L2) que elimina la primera aparición de X en la lista L1, el resultado lo deja en L2.

4. [10 puntos] 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 que permita:

- Leer sus datos.
- Imprimir sus datos.
- Almacenar los datos en una lista.
- Imprimir la lista de empleados.

5. [20 puntos] Realice un programa en PROLOG que calcule las series:

$$\blacksquare \sum_{i=a}^n (4i - 2)^2$$

$$\blacksquare \sum_{i=a}^n 3^{2i}$$

$$\blacksquare \sum_{i=a}^n \frac{-2i}{2i^2 - 12i - 5}$$

6. [20 puntos] 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

7. [20 puntos] Construya en PROLOG un predicado Anagrama(L,E) que compare dos hileras representadas por listas e indique si son un anagrama (Es un anagrama si las 2 palabras contienen exactamente las mismas letras). Algunos ejemplos son:

Anagrama([H, O, L, A], [O, L, A, S]) -----> False

Anagrama([M, O, R, A], [R, O, M, A]) -----> True

Anagrama([P, R, A, G, A], [P, A, G, A, R]) -----> True



oldemar **rodríguez**
CONSULTOR en MINERÍA DE DATOS