



UNIVERSIDAD DE GUADALAJARA



Centro Universitario de Ciencias Exactas e Ingenierías

Departamento de Ciencias Computacionales

Seminario de Solución de Problemas de Sistemas Basados en Conocimiento

Practica 6

Profesor:

Valdes Lopez Julio Esteban

Sección:

D05

Fecha:

12/09/2022

Alumno:

Sandoval Padilla Fernando Cesar

Código:

215685409

Carrera:

Ingeniería informática

Practica 6

%Escribe en Prolog las siguientes consultas y muestra sus resultados.

WRITE

La orden WRITE imprime en pantalla la cadena de caracteres en código ASCII por ejemplo:

1?-write("Hola mundo").

2?-write(Hola mundo).

3?-write(hola_mundo).

4?-write(hola).

```
?- write("Hola mundo").
Hola mundo
true.

?- write(Hola mundo).
ERROR: Syntax error: Operator expected
ERROR: write(Hola
ERROR: ** here **
ERROR: mundo) .
?- write(hola_mundo).
hola_mundo
true.

?- write(hola)
|
hola
true.

?- ■
```

Responde. ¿En qué casos da error y por qué?

En el caso 2 porque ocurre un error de sintaxis, la sintaxis correcta lleva comillas. En los demás casos es valido porque se les considera como una variable.

READ

La orden read sirve para almacenar el valor a una variable, ejemplo:

Escribe este enunciado en prolog y después de leer el nombre en minúsculas, el punto para terminar la sentencia y presiona enter.

5?-write("Nombre_"),read(NOMBRE),write("Tu nombre es: "), write(NOMBRE).

Ahora prueba sustituyendo write() por writeln(). Pega la consulta con sus resultados.

writeln("Nombre_"),read(NOMBRE),writeln("Tu nombre es: "), writeln(NOMBRE).

Seminario de Solución de Problemas de Sistemas Basados en Conocimiento

```
?- write("Nombre_"),read(NOMBRE),write("Tu nombre es: "), write(NOMBRE).
Nombre_fernando
|: .
Tu nombre es: fernando
NOMBRE = fernando.

?- writeIn("Nombre_"),read(NOMBRE),writeIn("Tu nombre es: "), writeIn(NOMBRE).
Correct to: "writeIn(\"Nombre_\")"? yes
Correct to: "writeIn(\"Tu nombre es: \")"?
Please answer 'y' or 'n'?
Please answer 'y' or 'n'? yes
Correct to: "writeIn(NOMBRE)"? yes
Nombre_
|: fernando
|: .
Tu nombre es:
fernando
NOMBRE = fernando.

?- ■
```

Otra forma sencilla de asignar un valor a una variable, colocando directamente el comando, seguido de la variable. Ejemplo:

6?-read(Color).

7?-read(Edad).

8?-read(edad).

Responde. Si es que hubiera un problema con las sentencias anteriores, ¿A qué se debe?

El problema es en el último, ya que las variables deben ser escritas con una mayúscula inicial y en este caso está en minúsculas.

```
?- read(Color).
|: blanco.

Color = blanco.

?- read(Edad).
|: 22.

Edad = 22.

?- read(edad).
|: 22.

false.
```

._?- ■

Escribe en un documento de prolog lo siguiente

leer(NOMBRE):-write("Nombre: "), read(NOMBRE), write("Hola "), write(NOMBRE).

Practica 6

leer:-write("Nombre: "), read(NOMBRE), write("Hola "), write(NOMBRE).

Guarda, compila y escribe las siguientes consultas.

9?- leer(X).

10?- leer.

```
?- leer(X).  
Nombre: fernando.  
Hola fernando  
X = fernando.  
  
?- leer.  
Nombre: fernando.  
Hola fernando  
true.  
?- ■
```

PLUS

Plus sirve para sumar los argumentos recibidos, ejemplo:

11?-plus(1,2,3).

12?-plus(1,2,SUMA).

13?-plus(1,2,5).

```
?- plus(1,2,3).  
true.  
  
?- plus(1,2,SUMA).  
SUMA = 3.  
  
?- plus(1,2,5).  
false.  
?- ■
```

Responde:

¿Qué argumento es el que tiene la suma de los otros elementos?

SUMA

¿Cuántos son los números que se pueden sumar?

2

BETWEEN

Encuentra un número en un rango, ejemplo:

14?-between(1,7,5).

15?-between(1,7,X).

Seminario de Solución de Problemas de Sistemas Basados en Conocimiento

16?-between(1,7,9).

```
?- between(1,7,5).  
true.
```

```
?- between(1,7,X).  
X = 1 ;  
X = 2 ;  
X = 3 ;  
X = 4 ;  
X = 5 ;  
X = 6 ;  
X = 7 .
```

```
?- between(1,7,9).  
false.
```

```
?-  
_____
```

¿Qué significa cada uno de los argumentos?

- El primero nos dice que efectivamente el numero 5 se encuentra entre el 1 y el 7
- El segundo nos dice todos los posibles números que pueden estar entre el 1 y el 7
- El tercero nos dice que el numero 9 no se encuentra entre el 1 y el 7

SUCC

Devuelve verdadero si el 2º argumento es $= 1^0 + 1$ y si el $1^0 \geq 0$,

ejemplo:

16?-succ(5,4).

17?-succ(5,5).

18?-succ(9,10).

19?-succ(0,1).

Escribe la consulta y resultado donde regrese falso la función succ\2

```
?- succ(5,4).  
false.
```

```
?- succ(5,5).  
false.
```

```
?- succ(9,10).  
true.
```

```
?- succ(0,1).  
true.
```

```
?-  
_____
```

IS

Es un predicado que define una expresión, ejemplo:

Practica 6

20?-4 is 2+2.

21?-4 is 2+3.

22?-5*2 is 10.

23?-10 is 5*2.

```
?- 4 is 2+2.  
true.
```

```
?- 4 is 2+3.  
false.
```

```
?- 5*2 is 10.  
false.
```

```
?- 10 is 5*2.  
true.
```

```
?- ■
```

Aritmética.

Prolog también incorpora funciones aritméticas

Realizar una consulta por cada una de estas funciones.

- abs

```
?- X is abs(-9).  
X = 9.
```

- sign

```
?- X is sign(-9).  
X = -1.  
~ ~ ~ ~ ~
```

- min

```
?- A is min(nan, 1).  
A = 1.
```

- max

```
?- X is max(5,10).  
X = 10.  
~
```

- random

```
?- X is random(0xFFFF).  
X = 16808.
```

- round

Seminario de Solución de Problemas de Sistemas Basados en Conocimiento

```
?- X is round(-1.0Inf).  
X = -1.0Inf.
```

- integer

```
?- X is integer(-20).  
X = -20.
```

- float

```
?- X is float(968).  
X = 968.0.
```

- sqrt

```
?- X is sqrt(50).  
X = 7.0710678118654755.
```

- sin

```
?- X is sin(50).  
X = -0.26237485370392877.
```

- cos

```
?- X is cos(50).  
X = 0.9649660284921133.
```

- tan

```
?- X is tan(50).  
X = -0.27190061199763077.  
-
```

- log

```
?- X is log(10).  
X = 2.302585092994046.
```

- log10

```
?- X is log10(10).  
X = 1.0.
```

- exp

Practica 6

```
?- X is exp(10).  
X = 22026.465794806718.
```

Escribe las siguientes reglas y pruébalas.

sumar_3_y_duplicar(X,Y):-Y is (X+3)*2.

sumar(Y):-Y is (10+3)*2.

```
?- sumar_3_y_duplicar(5,Y).  
Y = 16.
```

```
?- sumar(Y).  
Y = 26.
```

```
?-
```

```
?- sumar(26).  
true.
```

```
?- ■
```

Además de probar las reglas anteriores, muestra una consulta donde la regla sumar regrese falso.

```
?- sumar(5).  
false.
```

Funciones para trabajar con listas.

Ejecuta las siguientes consultas y escribe para que sirve cada una de ellas.

24?-reverse([1,2,3,4],L2).

25?-reverse([a,b,c,d],L2).

```
?- reverse([1,2,3,4],L2).  
L2 = [4, 3, 2, 1].
```

```
?- reverse([a,b,c,d],L2).  
L2 = [d, c, b, a].
```

```
?- ■
```

Invierten el orden de las listas

27?-length([1,2,3], L).

```
?- length([1,2,3], L).  
L = 3.
```

```
?- ■
```

Nos muestra la longitud de la lista, o el numero de elementos que hay en ella.

28?-sort([1,7,3], L2).

29?-sort([6,19,1,5], L2).

Seminario de Solución de Problemas de Sistemas Basados en Conocimiento

```
?- sort([1,7,3], L2).  
L2 = [1, 3, 7].
```

```
?- sort([6,19,1,5], L2).  
L2 = [1, 5, 6, 19].
```

```
?-
```

Ordenan los elementos de la lista de menor a mayor

```
30?-nextto(4,N,[1,7,3]).
```

```
31?-nextto(7,N,[1,7,3]).
```

```
32?-nextto(1,N,[1,7,3]).
```

```
?- nextto(4,N,[1,7,3]).  
false.
```

```
?- nextto(7,N,[1,7,3]).  
N = 3 ;  
false.
```

```
?- nextto(1,N,[1,7,3]).  
N = 7 ;  
false.
```

```
?- ■
```

Obtiene el siguiente si este se encuentra en la lista.