

1. Definir la relación `maximo(+X, +Y, ?Z)` que se verifique si Z es el máximo de X e Y. Por ejemplo:

```
?- maximo(2, 3, X) .
X=3
?- maximo(3, 2, X) .
X=3
```

2. La relación `primero(?L, ?X)` se verifica si X es el primer elemento de la lista L. Por ejemplo:

```
?- primero([a,b,c], X) .
X=a
```

- (a) Definir la relación en PROLOG.
- (b) Obtener la respuesta a las siguientes consultas:
 - I. `?- primero([X, b, c], a).`
 - II. `?- primero([X, Y], a).`
 - III. `?- primero(X, a).`

3. La relación `cola(?L1, ?L2)` se verifica si L2 es la lista obtenida a partir de la lista L1 suprimiendo el primer elemento. Por ejemplo:

```
?- cola([a,b,c], L) .
L=[b,c]
```

- (a) Definir la relación en PROLOG.
- (b) Obtener la respuesta a las siguientes consultas:
 - I. `?- cola([a|L], [b, c]).`
 - II. `?- cola(L, [b, c]).`

4. La relación `cons(?X, ?L1, ?L2)` se verifica si L2 es la lista obtenida añadiéndole X a L1 como primer elemento. Por ejemplo:

```
?- cons(a, [b,c], L) .
L = [a,b,c]
```

- (a) Definir la relación en PROLOG.
- (b) Obtener la respuesta a las siguientes consultas:
 - I. `?- cons(X, [b, c], [a, b, c]).`
 - II. `?- cons(a, L, [a, b, c]).`
 - III. `?- cons(b, L, [a, b, c]).`
 - IV. `?- cons(X, L, [a, b, c]).`

5. Definir la relación `suma_lista(+L, ?X)` que se verifique si X es la suma de los elementos de la lista de numeros L. Por ejemplo,

```
?- suma_lista([1, 3, 5], X) .
X=9
```

6. La relación `pertenece(?X, ?L)` se verifica si X es un elemento de la lista L. Por ejemplo:

```
?- pertenece(b, [a,b,c]).  
Yes  
?- pertenece(d, [a,b,c]).  
No
```

- (a) Definir la relación en PROLOG.
 - (b) Hacer el árbol de resolución para `?- pertenece(b, [a, b, c]).`
 - (c) Generar las consultas que permitan utilizar la relación definida para responder a las siguientes preguntas:
 - I. ¿Es c un elemento de [a, c, b, c]?
 - II. ¿Cuáles son los elementos de [a, b, a] ?
 - III. ¿Cuáles son los elementos comunes de [a, b, c], y [d, c, b]?
7. La relación `conc(?L1,?L2,?L3)` se verifica si L3 es la lista obtenida escribiendo los elementos L2 a continuación de los elementos de L1. Por ejemplo

```
?- conc([a,b],[c,d,e],L).  
L = [a,b,c,d,e]
```

- (a) Definir la relación en PROLOG.
 - (b) Hacer el árbol de resolución para `?- conc([a, b], [c, d, e], L).`
 - (c) Generar las consultas que permitan utilizar la relación definida para responder a las siguientes preguntas:
 - I. ¿Qué lista hay que añadirle a la lista [a, b] para obtener [a, b, c, d]?
 - II. ¿Qué listas hay que concatenar para obtener [a, b]?
 - III. ¿Pertenece b a la lista [a, b, c]?
 - IV. ¿Es [b, c] una sublista de [a, b, c, d]?
 - V. ¿Es [b, d] una sublista de [a, b, c, d]?
 - VI. ¿Cuál es el último elemento de [b, a, c, d]?
8. Un palíndromo es una palabra que se lee igual en los dos sentidos, por ejemplo *oso*. La relación `palindromo(+L)` se verifica si la lista L es un palíndromo. Por ejemplo

```
?- palindromo([o,s,o]).  
Yes  
?- palindromo([o,s,a]).  
No
```

- (a) Definir la relación en PROLOG.
 - (b) Hacer el árbol de resolución para
 - I. `?- palindromo([o, s, o]).`
 - II. `?- palindromo([o, s, a]).`
9. La relación `duplica(?L1,?L2)` que se verifique si L2 es la lista obtenida escribiendo cada elemento de L1 dos veces. Por ejemplo,

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

- (a) Definir la relación en PROLOG.
- (b) Hacer el árbol de resolución para `?- duplica([1, 2, 3], L).`

10. Definir la relación `cuentaN(+L, +N, ?LR)` que se verifica si LR es la cantidad de veces que se encuentra N en la lista L. Por ejemplo:

```
?- cuentaN([1, 2, 1, 2, 1], 1, 3) .
?- true.
?- cuentaN([1, 2, 1, 2, 1], 10, X) .
X=0.
?- cuentaN([1, 2, 1, 2, 1, 1, 1], 1, X) .
X=5.
```

11. Definir la relación `mayorN(+N, +L, ?LR)` que se verifica si LR es una lista con los elementos mayores que N encontrados en la lista L. Por ejemplo

```
?- mayorN(3, [1, 2, 3, 4, 5, 6], X) .
X = [4, 5, 6]
?- mayorN(3, [], X) .
X=[]
?- mayorN(3, [1, 2, 3, 4, 5, 6], [4, 5, 6]) .
true.
```

12. A continuación se presenta información de dos familias.

- En la primera familia,
 - el padre es Tomás García Pérez, nacido el 7 de Mayo de 1960, trabaja de profesor y gana 60 pesos diarios;
 - la madre es Ana López Ruiz, nacida el 10 de marzo de 1962, trabaja de médica y gana 90 pesos diarios;
 - el hijo es Juan García López, nacido el 5 de Enero de 1980, estudiante;
 - la hija es María García López, nacida el 12 de Abril de 1992, estudiante.
- En la segunda familia,
 - el padre es José Pérez Ruiz, nacido el 6 de Marzo de 1963, trabaja de pintor y gana 120 pesos diarios;
 - la madre es Luisa Gálvez Pérez, nacida el 12 de Mayo de 1964, trabaja de médica y gana 90 pesos diarios;
 - un hijo es Juan Luis Pérez Pérez, nacido el 5 de Febrero de 1990, estudiante;
 - una hija es María José Pérez Pérez, nacida el 12 de Junio de 1992, estudiante;
 - otro hijo es José María Pérez Pérez, nacido el 12 de Julio de 1994, estudiante.

Esta información puede representarse con cualquiera de las dos estructuras PROLOG que se muestran debajo.

```
familia(
  persona([tomas, garcia, perez], fecha(7, mayo, 1960), trabajo(profesor, 60)),
  persona([ana, lopez, ruiz], fecha(10, marzo, 1962), trabajo(medica, 90)),
  [persona([juan, garcia, lopez], fecha(5, enero, 1990), estudiante),
   persona([maria, garcia, lopez], fecha(12, abril, 1992), estudiante)]) .
familia(
  persona([jose, perez, ruiz], fecha(6, marzo, 1963), trabajo(pintor, 120)),
  persona([luisa, galvez, perez], fecha(12, mayo, 1964), trabajo(medica, 90)),
```

```
[persona([juan_luis,perez,perez], fecha(5,febrero,1990), estudiante),  
persona([maria_jose, perez, perez], fecha(12,junio,1992), estudiante),  
persona([jose_maria, perez, perez], fecha(12,julio,1994), estudiante)]].
```

Estructura 1

```
familia([  
persona([tomas,garcia,perez], fecha(7,mayo,1960), trabajo(profesor,60)),  
persona([ana,lopez,ruiz], fecha(10,marzo,1962), trabajo(medica,90)),  
persona([juan,garcia,lopez], fecha(5,enero,1990), estudiante),  
persona([maria,garcia,lopez], fecha(12,abril,1992), estudiante)]) .  
familia([  
persona([jose,perez,ruiz], fecha(6,marzo,1963),trabajo(pintor,120)),  
persona([luisa,galvez,perez], fecha(12,mayo,1964), trabajo(medica,90)),  
persona([juan_luis,perez,perez], fecha(5,febrero,1990), estudiante),  
persona([maria_jose,perez,perez], fecha(12,junio,1992), estudiante),  
persona([jose_maria,perez,perez], fecha(12,julio,1994), estudiante)]) .
```

Estructura 2

- (a) Discutir las ventajas /desventajas de cada una de estas estructuras con sus compañeros. Resumir en una tabla las conclusiones obtenidas.
- (b) Resolver los siguientes ejercicios utilizando la Estructura 1.
- I. Realizar las siguientes consultas:
 1. ¿existe familia sin hijos?
 2. ¿existe familia con un hijo?
 3. ¿existe familia con dos hijos?
 4. ¿existe familia con tres hijos?
 5. ¿existe familia con cuatro hijos.?
 - II. Buscar los nombres de los padres de familia con tres hijos.
 - III. Definir la relación casada(X) que se verifique si X es una mujer casada.
 - IV. Preguntar por las mujeres casadas.
 - V. Determinar el nombre de todas las mujeres casadas que trabajan.
 - VI. Definir la relación persona(X) que se verifique si X es una persona existente en la base de datos.
 - VII. Preguntar por los nombres y apellidos de todas las personas existentes en la base de datos.
 - VIII. Definir la relación sueldo(X, Y) que se verifique si el sueldo de la persona X es Y.
 - IX. Definir la relación total(L, Y) de forma que si L es una lista de personas, entonces Y es la suma de los sueldos de las personas de la lista L.