

5. Sea el siguiente enunciado:

Bertoldo y Bartolo son rufianes.
Romeo y Bertoldo, como su nombre indica, son nobles.
Bartolo es un plebeyo.
Gertrudis y Julieta son damas.
Julieta es hermosa.
Los plebeyos desean a cualquier dama, mientras que los nobles
solo a aquellas que son hermosas.
Los rufianes, para satisfacer sus instintos, raptan a las personas
a las que desean.

Representa el enunciado como un programa Prolog y utiliza el programa
para responder las siguientes preguntas:

- ¿Qué noble es un rufián?
- ¿A quién podría raptar Romeo?
- ¿Quién puede raptar a Julieta?
- ¿Quién rapta a quién?
- ¿A quién desea Bartolo?
- ¿Y Romeo?
- ¿Qué hermosa dama es deseada por Bartolo?

6. Definir la relación palindromo(L) que verifique si una lista L es un
palíndromo. Por ejemplo,

```
?- palindromo([o,s,o]).  
yes
```

7. Supongamos que representamos los puntos del plano mediante términos de la forma `punto(X,Y)` donde X e Y son números, y los segmentos del plano mediante términos de la forma `segmento(P1,P2)` donde $P1$ y $P2$ son los puntos extremos del segmento. Define las relaciones `vertical(S)` y `horizontal(S)` que se verifiquen si S es un segmento vertical u horizontal, respectivamente. Por ejemplo,

```
?- vertical(segmento(punto(1,2),punto(1,3))).
yes
?- vertical(segmento(punto(1,2),punto(4,2))).
no
?- horizontal(segmento(punto(1,2),punto(5,2))).
yes
```

8. Define una relación $f(X,Y)$ de forma que:
- si $X < 3$, entonces $Y=0$.
 - si $3 \leq X < 6$, entonces $Y=2$
 - si $6 \leq X$, entonces $Y=4$
- y construye el árbol de derivaciones para los objetivos
- ```
?- f(1,Y), 2 < Y.
?- f(7,Y).
```

9. ¿Qué hace el siguiente programa Prolog?

```
nose([]).
nose([_]).
nose([X,X|L]):-nose([X|L]).
```

10. Escribe un programa Prolog que compruebe si la longitud de una lista es impar.

11. Escribe un programa Prolog que compruebe si la longitud de una lista es par.

12. Escribe una nueva versión de los dos problemas anteriores por recursión cruzada.
  
  
  
  
  
  
  
  
  
  
13. Escribe un programa en Prolog que elimine las repeticiones de elementos de una lista, guardando el resultado en una segunda lista. Por ejemplo, `[a,b,h,j]` se obtendría como resultado de eliminar los elementos repetidos de la lista `[a,b,h,b,a,j]`.
  
  
  
  
  
  
  
  
  
  
14. Escribe un programa que cuente el número de veces que un elemento se encuentra repetido en una lista.
  
  
  
  
  
  
  
  
  
  
15. Escribe un programa en Prolog para calcular el mayor elemento de una lista de enteros.
  
  
  
  
  
  
  
  
  
  
16. Escribe un programa en Prolog para dividir una lista respecto un umbral, dejando los valores menores a la izquierda y los mayores a la derecha del umbral. Por ejemplo, el resultado de partir la lista `[2,7,4,8,9,1]` respecto al umbral 6 serían las listas `[2,4,1]` y `[7,8,9]`.
  
  
  
  
  
  
  
  
  
  
17. Escribe un programa en Prolog para sumar los elementos de una lista de enteros.