**Ejercicio 1:** El predicado pertenece comprueba si el elemento x pertenece a una lista.

pertenece(X,[X|\_]).

pertenece(X,[\_|L2]):- pertenece(X,L2).

Lo hemos realizando tomando como case base que el elemento que queremos comprobar es el primero de la lista, y como caso recursivo volver a evaluar con el resto de la lista.

**Ejercicio 2:** El predicado invierte comprueba si el segundo argumento que es una lista es la inversa que el primer argumento, que también es una lista.

invierte([],[]).

invierte([H|T],L):- invierte(T,R), concatena(R,[H],L).

Hemos realizado este predicado tomando como caso base dos listas vacías, que evidentemente se cumple que sean la inversa la una de la otra.

En el caso recursivo llamamos al propio predicado con el resto de la lista inicial, y de segundo argumento el resultado de concatenar el primer elemento de la lista original con la lista invertida.

**Ejercicio 3:**

**Ejercicio 4:**

**Ejercicio 5:** Este predicado se encarga de extraer un elemento dado de una lista, y en el tercer argumento va la lista resultado de la extracción

extract(L1,X,L2):- concatena(L4,L5,L1),

concatena([X],L3,L5),

concatena(L4,L3,L2).

Lo hemos realizado haciendo uso del predicado concatena.

**Ejercicio 6:** Estos predicados son iguales, simplemente comprueban si el elemento X está contenido en la lista L.

man\_pref(X,L) :- pertenece(X,L).

woman\_pref(X,L) :- pertenece(X,L).

**Ejercicio 7:**

**Ejercicio 8:**