Ejercicio 4.1. Definir la relación `diferencia(C1,C2,C3)` que se verifica si `C3` es la diferencia de los conjuntos `C1` y `C2`. Por ejemplo,

```
?- diferencia([a,b],[b,c],X)
X = [a] ;
No
```

Hacer una versión con negación y otra con corte.

Ejercicio 4.2. Definir la relación `agregar(X,L,L1)` que se verifica si `L1` es la lista obtenida añadiéndole `X` a `L`, si `X` no pertenece a `L` y es `L` en caso contrario. Por ejemplo,

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

Ejercicio 4.3. Definir el predicado `separa(L1,L2,L3)` que separa la lista de números `L1` en dos listas: `L2` formada por los números positivos y `L3` formada por los números negativos o cero. Por ejemplo,

```
?- separa([2,0,-3,5,0,2],L2,L3).
L2 = [2, 5, 2]
L3 = [0, -3, 0]
Yes
```

Proponer dos soluciones, una sin corte y otra con corte.

Ejercicio 4.4. Definir el predicado `suma_pares(L,N)` que se verifica si `N` es la suma de todos los números pares de la lista `L`. Por ejemplo,

```
?- suma_pares([2,3,4],N).
N = 6 ;
No
```

Hacer una versión con negación y otra con corte.

Ejercicio 4.5. Definir el predicado `exponente_de_dos(N,E)` que se verifica si `E` es el exponente de 2 en la descomposición de `N` como producto de factores primos. Por ejemplo,

```
?- exponente_de_dos(40,E).
E=3
?- exponente_de_dos(49,E).
E=0
```

Hacer una versión con negación y otra con corte.

Ejercicio 4.6. Definir la relación `lista_a_conjunto(+L,-C)` que se verifique si `C` es el conjunto correspondiente a la lista `L` (es decir, `C` contiene los mismos elementos que `L` en el mismo orden, pero si `L` tiene elementos repetidos sólo se incluye en `C` la última aparición de cada elemento). Por ejemplo,

```
?- lista_a_conjunto([b,a,b,d],C).
C = [a, b, d]
```

Nota: La relación `lista_a_conjunto` se corresponde con la relación definida `list_to_set`.

Ejercicio 4.7. Definir la relación `crecimientos(+L1,-L2)` que se verifique si `L2` es la lista correspondiente a los crecimientos de la lista numérica `L1`; es decir, entre cada par de elementos consecutivos `X` e `Y` de `L1` coloca el signo `+` si $X < Y$ e y signo `-` en caso contrario. Por ejemplo,

```
?- crecimientos([1,3,2,2,5,3],L).
L = [1, +, 3, -, 2, -, 2, +, 5, -]
```

Dar una definición sin corte y otra con corte.

Ejercicio 4.8. Definir las siguientes relaciones:

- `menor_divisor_propio(+N,?X)` que se verifique si `X` es el menor divisor de `N` mayor o igual que 2. Por ejemplo,

```
?- menor_divisor_propio(30,X).
X = 2
?- menor_divisor_propio(3,X).
X = 3
```

- `factorización(+N,-L)` que se verifique si `L` es la lista correspondiente a la descomposición del número `N` en factores primos (se considera los que elementos de `L` están ordenados de manera creciente). Por ejemplo,

```
?- factorización(12,L).
L = [2, 2, 3] ;
No
?- factorización(1,L).
L = [] ;
No
```

Ejercicio 4.9. Definir la relación `calcula(+N,+M,?X)` que se verifique si `X` es el menor múltiplo de `N` tal que la suma de sus dígitos es mayor que `M`. Por ejemplo,

```
?- calcula(3,10,X).
X = 39
Yes
?- calcula(7,20,X).
X = 399
Yes
```

Ejercicio 4.10. Un número es libre de cuadrados si no es divisible por el cuadrado de ningún número mayor que 1. Definir la relación `libre_de_cuadrados(+N)` que se verifique si el número `N` es libre de cuadrados. Por ejemplo,

```
?- libre_de_cuadrados(30).
Yes
?- libre_de_cuadrados(12).
No
```

Ejercicio 4.11. Definir la relación `suma_libres_de_cuadrados(+L,-S)` que se verifique si `S` es la suma de los números libres de cuadrados la lista numérica `L`. Por ejemplo,

```
?- suma_libres_de_cuadrados([6,12,18,30],S).  
S = 36
```

Nota: Dar dos definiciones, una con negación y otra con corte.

Ejercicio 4.12. Definir la relación `max_lista(+L,?X)` que se verifique si `X` es el máximo de la lista de números `L`. Por ejemplo,

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

Ejercicio 4.13. Definir la relación `longitud_scm(+L1,+L2,-N)` que se verifique si `N` es la longitud de las subsucesiones comunes maximales de las listas `L1` y `L2`. Por ejemplo,

```
?- longitud_scm([2,1,4,5,2,3,5,2,4,3],[1,7,5,3,2],N).  
N = 4 ;  
No
```

ya que `[1,5,3,2]` es una subsucesión de las dos listas y no poseen ninguna otra subsucesión común de mayor longitud. Obsérvese que los elementos de la subsucesión no son necesariamente elementos adyacentes en las listas.