Lógico - Listas - Recursividad

a. `encolar(E,L,LConE)`, relaciona un elemento con una lista y la lista que resulta de agregar el elemento al final. Ej:

```
? – encolar(7, [3,4,5],[3,4,5,7]).
true.
```

b. `maximo(L, Max)`, relaciona una lista con el elemento más grande. Realizarlo con y sin recursividad.

c. `unirSinRepeticiones([3,4,5],[9,4,10],[3,5,9,4,10])`. Relaciona 2 listas con la lista que se obtiene de la unión de estas sin elementos repeticiones

d. `inteseccion([3,4,5],[2,7,5],[5])`. Relaciona la lista 1 con la lista 2 y la lista 3 que es la resultante
de la intersección de las otras 2.

e. `esCreciente([3,4,5,7])`. Conocer si todos los elementos de la lista están ordenados de menor a mayor.

f. `sublistaMayoresA(L, Elem, Mayores)`, relaciona una lista con un elemento y las sublistas cuyos elementos sean mayores a Elem.

```
? - mayoresA([3,4,5,6], 4, L)
L = [5, 6] ;
L = [5] ;
L = [6] ;
L = [].
```

g. `reversa([4,9,3],[3,9,4])`. Relaciona una lista con su reversa.

Explosión Combinatoria:

Desarrollar el predicado **entretenimientos/2**, relaciona una cantidad de dinero con los entretenimientos posibles que puede realizar con dicha cantidad.

```
entretenimiento(cine).
entretenimiento(teatro).
entretenimiento(pool).
entretenimiento(parqueTematico).
costo(cine, 30).
costo(teatro, 40).
costo(pool, 15).
costo(parqueTematico, 50).

?- entretenimientos(100, ListaEntre).
ListaEntre = [cine, teatro, pool] ;
ListaEntre = [cine, teatro] ;
ListaEntre = [cine, pool, parqueTematico] ;
ListaEntre = [cine, pool] ;
ListaEntre = [cine, parqueTematico] ;
ListaEntre = [cine] ;
ListaEntre = [teatro, pool] ;
ListaEntre = [teatro, parqueTematico] ;
ListaEntre = [teatro] ;
ListaEntre = [pool, parqueTematico] ;
ListaEntre = [pool] ;
ListaEntre = [parqueTematico] ;
ListaEntre = [].
```