

EJERCICIOS DE PROLOG

1. Programar un predicado `suma_parcial(C,SP,SC)` que determine qué números del conjunto `C` suman la cantidad `SP`. Esto podría servir para devolver el cambio en monedas de una cantidad `SP` determinada. Así por ejemplo:

```
?- suma_parcial([1,2,2,3,5], 5, SC).
```

```
SC = [1,2,2];
```

```
SC = [2,3];
```

```
SC = [2,3];
```

```
SC = [5];
```

2. Programar un predicado en prolog `salto(X/Y,X1/Y1)` que compute el salto de un caballo de ajedrez, teniendo en cuenta que las coordenadas de partida `X/Y` van a estar siempre instanciadas
3. Programar un predicado `saltos_caballo(Camino)` que nos de un posible camino del caballo. Queremos hacer preguntas del tipo: “el caballo comienza en 2/1 y queremos saber si puede llegar a la octava fila en 4 movimientos y sabiendo por qué casillas pasamos”

```
?- saltos_caballo([2/1, R, 5/4, S, X/8]).
```

4. Borra **todas** las ocurrencias de un elemento. Hacer dos versiones más, una con corte verde y otra con corte rojo.

```
?- borra([1, 2, 1, 3, 1], 1, Y).
```

```
Y=[2,3]
```

5. Borra una sola ocurrencia:

```
?- borra1([1, 2, 1, 3, 1], 1, Y).
```

```
Y = [2, 1, 3, 1]
```

6. Evitar el “backtracking” en el ejercicio anterior donde sea necesario.
7. Calcular el mínimo valor de una lista: `minlist(L,M)`
8. Comprueba si una lista de elementos está ordenada `ordenados([1,2,3,4])`
9. Ordena una lista de elementos:

```
ordena([3,1,5],L).
```

```
L=[1,3,5]
```

10. Comprobar si una lista es palíndroma:

```
?- palindrome([1,2,3,3,2,1]).
```

```
yes
```

```
?- palindrome([1,2,3,X,Y,1]).
```

```
X = 3
```

```
Y = 2
```

11. Rotar una lista:

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

```
L=[2,3,1]
```