

Apellidos:

Nombre:

Ejercicio 1 [2.5 puntos] Considera el siguiente programa

```
misterio(X,N,L):-  
    length(L,N),  
    append([X],L1,L),  
    append(L2,[X],L),  
    L1 = L2.
```

que recibe como dato de entrada una constante X y un número natural N y devuelve la lista L .

1. Describe el comportamiento general de `misterio`.
2. Define un predicado *recursivo* `crealista(X,N,L)` que a partir de los mismos datos de entrada X y N devuelva la misma lista L que `misterio`.

Ayuda:

```
?- length(L,5).  
    L = [A1, A2, A3, A4, A5] ;  
    No
```

donde $A1, A2, A3, A4, A5$ son cinco variables distintas.

Ejercicio 2 [2.5 puntos] Definir el predicado `listacero(L1,L2)` que tome como entrada una lista de números $L1$ y devuelva una lista no vacía $L2$ formada por elementos de $L1$, en el mismo orden, verificando la condición de que los elementos de $L2$ suman cero. Por ejemplo,

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

Ejercicio 3 [2.5 puntos] Definir el predicado `n_apply(+Term,+Lista)` que se verifica si `Term` es demostrable después de aumentar el número de sus argumentos con los elementos de `Lista`. Por ejemplo

```
?- n_apply(append([1,2]),[X,[1,2,3,4,5]]).  
    X = [3, 4, 5] ;  
    No
```

Ejercicio 4 [2.5 puntos] Definir el predicado `agrupa(L1,L2)` que tome como entrada la lista $L1$ y devuelva la lista $L2$, que es la lista $L1$ en la que hemos reordenado los elementos para que los que sean iguales estén juntos. Por ejemplo,

```
?- agrupa([a,b,a,c,c,a,b,c,b],L).  
    L = [a, a, a, b, b, b, c, c, c] ;  
    No
```