

**Apellidos:**

**Nombre:**

*Notas*

1. *En la evaluación de los ejercicios se tendrá en cuenta la corrección, simplicidad y eficiencia de la respuesta.*
2. *Hay que describir las definiciones auxiliares (menos las del sistema).*

**Ejercicio 1** [2 puntos] Definir la relación `mayor_suma_argumentos(+L, -N)` que se verifique si N es la mayor suma de los argumentos de la lista L. Por ejemplo,

```
?- mayor_suma_argumentos([g(7),f(1,a,2),g(7,a,5),f(b,c)], N).
N = 12
```

**Solución:**

```
mayor_suma_argumentos(L,N) :-
    member(T1,L),
    suma_argumentos(T1,N),
    not((member(T2,L), suma_argumentos(T2,M), M>N)).

suma_argumentos(T,N) :-
    T =.. [_|L],
    suma_lista(L,N).

suma_lista([],0).
suma_lista([X|L],N) :-
    number(X), !,
    suma_lista(L,M),
    N is X+M.
suma_lista([X|L],N) :-
    % not(number(X)),
    suma_lista(L,N).
```

**Ejercicio 2** [2 puntos] Definir la relación `minimos(+L1, -L2)` que se verifique si L2 es la lista obtenida sustituyendo cada elemento de la lista numérica no vacía L1 por el menor elemento de L1. Por ejemplo,

```
?- minimos([4,2,1,5,2,4],L).
L = [1, 1, 1, 1, 1, 1]
```

Para aumentar la eficiencia, se debe de recorrer la lista L1 el menor número de veces posible.

**Solución:**

```

minimos([X],[X]).
minimos([X,Y|L1],[U,U|L2]) :-
    Z is min(X,Y),
    minimos([Z|L1],[U|L2]).

```

**Ejercicio 3** [3 puntos] Definir la relación `codigo(+L1,-L2)` que se verifique si `L2` es la lista obtenida sustituyendo cada letra de la lista de letras `L1` por el orden correspondiente de dicha letra en la ordenación de la lista `L1`. Por ejemplo,

```

?- codigo([p,r,o,l,o,g],X).
X = [4, 5, 3, 2, 3, 1]
?- codigo([p,a,r,r,a],X).
X = [2, 1, 3, 3, 1]

```

.....

**Solución:**

```

codigo(L1,L2) :-
    sort(L1,L3),
    codigo_aux(L1,L3,L2).

```

`codigo_aux(+L1,+L3,-L2)` se verifica si `L2` es la lista sustituyendo cada elemento de `L1` por su posición en `L3`. Su definición es

```

codigo_aux([],_,[]).
codigo_aux([X|L1],L3,[N|L2]) :-
    n_nth1(N,L3,X),
    codigo_aux(L1,L3,L2).

```

`n_nth1(-N,+L,+X)` se verifica si `X` es el elemento que ocupa la posición `N` en la lista `L`. Corresponde a la definida del sistema `nth1`. Su definición es

```

n_nth1(1,[X|_],X) :- !.
n_nth1(N,[_|L],Y) :-
    n_nth1(M,L,Y),
    N is M+1.

```

**Segunda parte** [3 puntos] En el laboratorio.