

---

PRIMERA PARTE: (7 PUNTOS)

---

**Ejercicio 1** [1.5 puntos] Definir el predicado `herbrand(+L1,-L2)` que reciba como dato de entrada una lista `L1` no vacía de fórmulas atómicas cerradas de un lenguaje sin símbolos de función y devuelva en lista `L2` el universo de Herbrand del lenguaje determinado por esas fórmulas (esto es, las constantes que aparecen como argumentos de las fórmulas). Por ejemplo

```
?- herbrand([padre(juan,luis),padre(antonio,jaime),
            madre(ana,luis),soltero(emilio),
            primo(anselmo,ana),
            hermanos(silvia,juan,jose)],L).
```

```
L = [ana, anselmo, antonio, emilio, jaime, jose, juan, luis, silvia]
No
```

---

**Ejercicio 2** [1 punto] Consideremos que tenemos las siguientes constantes ordenadas de la siguiente manera

```
orden([q,w,e,r,t,y,u,i,o,p,a,s,d,f,g,h,j,k,l,z,x,c,v,b,n,m]).
```

Se pide el completar el predicado `n_ord(L1,L2)` y definir los predicados auxiliares correspondientes de manera que `n_ord` reciba como dato de entrada una lista `L1` de constantes (para las que está definido el orden) y devuelva la lista ordenada `L2`.

```
n_ord([], ... ).
n_ord([X|R], L2) :-
    divide(X, ... ),
    n_ord(... ),
    n_ord(... , ... ),
    append(... , [X| ... ],L2).
```

Por ejemplo,

```
?- n_ord([m, m, a, a, z, z, y, m, t, r, e, w, q],L).
L = [m, m, m, z, z, a, a, y, t, r, e, w, q]
Yes
```

---

Apellidos:  
Nombre:

**Ejercicio 3** [1.5 puntos] Definir un predicado `profundidad(L1,L2)` que tome como entrada una lista `L1` y devuelva una lista `L2` formada a partir de la lista `L1` donde hemos sustituido cada elemento por la profundidad a la que se encuentra. Consideraremos que la lista vacía nunca es elemento de otra lista en el dato de entrada. Por ejemplo:

```
?- profundidad([a,b,[c,[f,g],e],j,k],L).  
L = [0, 0, [1, [2, 2], 1], 0, 0] ;  
No
```

---

**Ejercicio 4** [1.5 puntos] Considera el siguiente predicado `misterio_1` que toma como dato de entrada una lista `L1` no vacía

```
misterio_1(L1,L2) :-  
    misterio_aux(L1, [],L2).  
  
misterio_aux([X],R,[X|R]).  
misterio_aux([X|L1],Acum,[X|L2]) :-  
    misterio_aux(L1,[X|Acum],L2).
```

1. Dibuja el árbol de resolución correspondiente a la pregunta `?-misterio_1([a,b,c],Z)`
2. Describe el comportamiento del predicado `misterio_1` en general.
3. Completa el siguiente predicado `misterio_2(L1,L2)` de forma que se mantenga la misma relación entre `L1` y `L2`.

```
misterio_2(L1,L2) :-  
    append(...),  
    reverse(...),  
    append(...).
```

---

**Ejercicio 5** [1.5 puntos] Escribir un programa de 3 cláusulas y una pregunta para ese programa de forma que si ordenamos las cláusulas de distinta manera podemos obtener ninguna, una o infinitas respuestas.

---

SEGUNDA PARTE: (3 PUNTOS)

---

La segunda parte del examen se realizará en los laboratorios.

---