

Problema A: Dadas cinco personas, que tienen cinco casas de colores diferentes, y cinco profesiones, animales, bebidas y nacionalidades diferentes, y sabiendo que:

- 1 - El que vive en la casa roja es de Perú
- 2 - Al francés le gusta el perro
- 3 - El pintor es japonés
- 4 - Al chino le gusta el ron
- 5 - El húngaro vive en la primera casa
- 6 - Al de la casa verde le gusta el coñac
- 7 - La casa verde está justo a la izquierda de la blanca
- 8 - El escultor cría caracoles
- 9 - El de la casa amarilla es actor
- 10 - El de la tercera casa bebe cava
- 11 - El que vive al lado del actor tiene un caballo
- 12 - El húngaro vive al lado de la casa azul
- 13 - Al notario le gusta el whisky
- 14 - El que vive al lado del médico tiene un ardilla,

escribe un programa Prolog que averigüe para cada persona todas sus características de la forma [numcasa,color,profesión,animal,bebida,pais] averiguables.

Nota: partiendo de una solución [[1,A1,B1,C1,D1,E1] , . . . , [5,A5,B5,C5,D5,E5]], se pueden imponer todas las condiciones sobre ésta con **member** o similares.

Problema B:

Adapta (es obligatorio) el esquema Prolog de abajo para resolver los tres problemas. Piensa cómo representar los estados, y qué pasos entre estados hay. Por ejemplo, en el problema de los cubos de agua, el estado inicial puede ser [0,0], y el final [0,4].

B.1. Hacer aguas: disponemos de un grifo de agua, un cubo de 5 litros y otro de 8 litros. Se puede verter el contenido de un cubo en otro (hasta vaciar el primero, o hasta llenar el otro), llenar un cubo, o vaciar un cubo del todo. Escribir un programa Prolog que diga la secuencia más corta de operaciones para obtener exactamente 4 litros de agua en el cubo de 8 litros.

B.2. Buscamos la manera más rápida para tres misioneros y tres caníbales de cruzar un río en una canoa que puede ser utilizada por 1 ó 2 personas (misioneros o caníbales), pero siempre evitando que los misioneros queden en minoría en cualquier orilla (por razones obvias).

B.3. Trata de averiguar la manera más rápida que tienen cuatro personas P_1 , P_2 , P_5 y P_8 para cruzar de noche un puente que sólo aguanta el peso de dos, donde tienen una única e imprescindible linterna y cada P_i tarda i minutos en cruzar. Dos juntos tardan como el más lento de los dos.

```
main:- EstadoInicial = ...,      EstadoFinal   = ...,
    between(1,1000,CosteMax),      % Buscamos solución de coste 0; si no, de 1, etc.
    camino( CosteMax, EstadoInicial, EstadoFinal, [EstadoInicial], Camino ),
    reverse(Camino,Camino1), write(Camino1), write(' con coste '), write(CosteMax), nl, halt.

camino( 0, E,E, C,C ).              % Caso base: cuando el estado actual es el estado final.
camino( CosteMax, EstadoActual, EstadoFinal, CaminoHastaAhora, CaminoTotal ):-
    CosteMax>0,
    unPaso( CostePaso, EstadoActual, EstadoSiguiente ), % En B.1 y B.2, CostePaso es 1.
    \+member( EstadoSiguiente, CaminoHastaAhora ),
    CosteMax1 is CosteMax-CostePaso,
    camino(CosteMax1,EstadoSiguiente,EstadoFinal, [EstadoSiguiente|CaminoHastaAhora], CaminoTotal).

unPaso(....):- ...
...
```

Problema C:

Un programa escrito en el lenguaje *sumbol* tiene la siguiente sintaxis:

```
<programa>  -->  begin <instrucciones> end

<instrucciones>  -->  <instruccion>
<instrucciones>  -->  <instruccion> ; <instrucciones>

<instruccion>  -->  <variable> = <variable> + <variable>
<instruccion>  -->  if <variable> = <variable> then <instrucciones> else <instrucciones> endif
<variable>-->  x
<variable>-->  y
<variable>-->  z
```

Tres ejemplos de programas *sumbol*:

```
begin x=x+z end
begin x=x+y; z=z+z; x=y+x end
begin x=y+z; if z=z then x=x+z; y=y+z else z=z+y endif; x=x+z end
```

Se pide: escribir en Prolog un sencillo analizador sintáctico para el lenguaje *sumbol*, es decir, una cláusula `programa(P)` que se satisface si la lista de átomos `P` contiene un programa *sumbol* sintácticamente correcto, y que falla en caso contrario. Ejemplos:

```
?- programa( [begin, z, =, x, +, y, end] ).
yes
```

```
?- programa( [begin, z, =, x, +, y, ;, x, =, z, z, end] ).
no
```

(en el segundo ejemplo falta un "+"). Para ello, haz corresponder una cláusula a cada regla de la gramática.