

Problema A: Adapta el problema de “cifras” de la página 15 de la hoja p6 de los apuntes para que también use la división (sin dar nunca división por cero) y además encuentre las soluciones más cortas primero (las que menos elementos de la lista L usan).

Problema B:

B.1. Buscamos la manera más rápida para tres misioneros y tres caníbales de cruzar un río, disponiendo de 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.2. Dado un natural $n > 0$, una posición inicial $(Fila_I, Columna_I)$, una posición final $(Fila_F, Columna_F)$, y un número de pasos P , encontrar un camino de $(Fila_I, Columna_I)$ a $(Fila_F, Columna_F)$, en un tablero de ajedrez de $n \times n$ en exactamente P pasos de caballo. El programa ha de fallar si para la n en cuestión no existe tal camino.

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.

Adapta (es obligatorio) el siguiente esquema Prolog para resolver los tres problemas:

```
camino( E,E, C,C ).
camino( EstadoActual, EstadoFinal, CaminoHastaAhora, CaminoTotal ):-
    unPaso( EstadoActual, EstSiguiente ),
    \+member(EstSiguiente,CaminoHastaAhora),
    camino( EstSiguiente, EstadoFinal, [EstSiguiente|CaminoHastaAhora], CaminoTotal ).

solucionOptima:-
    nat(N),                % Buscamos solución de "coste" 0; si no, de 1, etc.
    camino(..., ..., ... , C), %
    length(C,N),           % Aquí el coste es la longitud de C (no siempre lo será).
    write(C).
```

Problema C: File `tsp.pl` contains a branch-and-bound solution to a Traveling-Salesman-like problem. Try to improve it as indicated in the file.