

PROLOG a la vida real

Enunciar i resoldre un problema qualsevol en PROLOG. Ha de ser un problema de lògica, de manera que PROLOG sigui el llenguatge més adequat per a la seva solució. L'entrega ha d'incloure:

- L'enunciat del problema
- La solució en PROLOG
- La solució o plantejament de la solució en un llenguatge procedural
- Les conclusions sobre la idoneïtat o no del PROLOG per a la resolució del problema plantejat en comparació amb la solució procedural.

Data máxima d'entrega: Dijous 19 de gener de 2012

No ha de ser cap problema dels vists a classe o els enunciats en el contingut del curs.

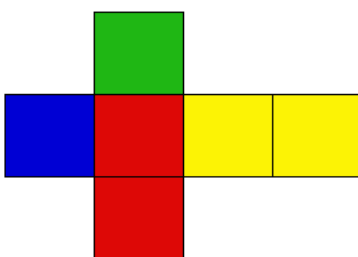
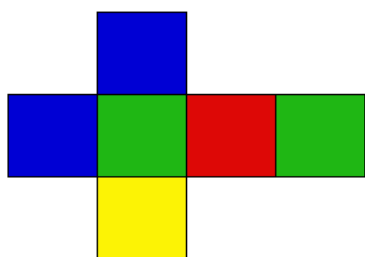
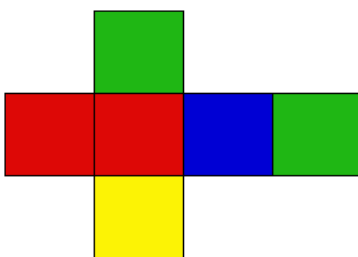
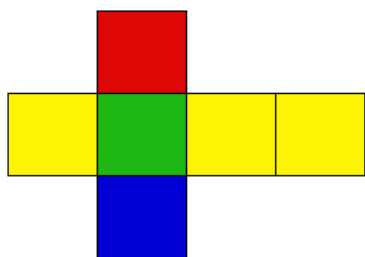
Exemple: Colorins

Passejant pel mercat de Santa Maria un diumenge al matí, vaig aturar-me a una parada dedicada a la venda de jocs de lògica, la gent s'aturava i jugava amb els joquets. A mí me va cridar especialment l'atenció un artilugi fet de fusta, quatre cubets amb les cares de colors un damunt l'altre, ficats dins una espècie de capsa, també de fusta, que deixava entreveure els colors de totes les cares. Aquell homonet me va veure interessat i amb una rialleta me va dir: aquest és especialment complex, s'han de posar els cubs de manera que a cada una de les quatre columnes hi hagi quatre colors diferents. Bé, vaig pensar jo, no deu ser tan difícil, i el vaig comprar. Una vegada a casa i abans de començar amb l'odissea de resoldre el trencaclosques, se'm va ocórrer una idea; si tant difícil era, seria un bon pretext per utilitzar el PROLOG.

A la setmana següent vaig tornar al mercat, i vaig tornar a aturar-me a la mateixa parada, que tornava a estar plena de gent. L'encarregat se'n va recordar de mí i em va dir irònicament: què, com va anar el trencaclosques de colorins. El vaig mirar amb una mitja rialla i li vaig contestar, bastant bé, de fet vaig trobar dues solucions distintes (i comptant totes les permutacions n'hi havia 192).

Escriviu un programa en PROLOG, que us permeti arribar a les mateixes solucions.

Dibuix dels cubs:



Solució en PROLOG:

```
/* veritat si tots els elements d'una llista són diferents */
```

```
diferents([]).
```

```
diferents([A|L]):-not(pertany(A,L)), diferents(L).
```

```
pertany(X,[X|_]).
```

```
pertany(X,[_|L]):-pertany(X,L).
```

```
/* color(cub, cara, color) */
```

```
color(1,1,grog).
```

```
color(1,2,verd).
```

```
color(1,3,grog).
```

```
color(1,4,grog).
```

```
color(1,5,blau).
```

```
color(1,6,vermell).
```

```
color(2,1,vermell).
```

```
color(2,2,vermell).
```

```
color(2,3,blau).
```

```
color(2,4,verd).
```

```
color(2,5,grog).
```

```
color(2,6,verd).
```

```
color(3,1,blau).
```

```
color(3,2,verd).
```

```
color(3,3,vermell).
```

```
color(3,4,verd).
```

```
color(3,5,grog).
```

```
color(3,6,blau).
```

```
color(4,1,blau).
```

```
color(4,2,vermell).
```

```
color(4,3,grog).
```

```
color(4,4,grog).
```

```
color(4,5,vermell).
```

```
color(4,6,verd).
```

```
/* Possibles maneres d'ordenar les cares d'un cub */
```

```
/* Es repeteixen els 3 primers elements per poder */
```

```
/* generar automàticament tots els possibles camins */
```

```
/* de 4 cares. */
```

```
/* Hi ha tres possibles posicions del cub però a la */
```

```
/* vegada poden estar cap a un costat o cap a l'altre */
```

```
/*
```

```
6
```

```
1 2 3 4    numeració de les cares del cub
```

```
5
```

```
*/
```

```
camí([1,2,3,4,1,2,3]).
```

```
camí([3,2,1,4,3,2,1]).
```

```
camí([2,6,4,5,2,6,4]).
```

```
camí([4,6,2,5,4,6,2]).
```

```
camí([1,6,3,5,1,6,3]).
```

```
camí([3,6,1,5,3,6,1]).
```

/* agafa seqüències de 4 cares per a cada cub */

agafa(L):-cami(L2),agafa4(L2,L).

agafa4([A,B,C,D|_],[A,B,C,D]).

agafa4(_|L1,L2):-agafa4(L1,L2).

/* Col·loca els colors a cada cara i cerca les solucions */

closca:-agafa([A1,A2,A3,A4]),agafa([B1,B2,B3,B4]),

color(1,A1,N1),color(2,B1,M1),N1\=M1,

color(1,A2,N2),color(2,B2,M2),N2\=M2,

color(1,A3,N3),color(2,B3,M3),N3\=M3,

color(1,A4,N4),color(2,B4,M4),N4\=M4,

agafa([C1,C2,C3,C4]),

color(3,C1,O1),

color(3,C2,O2),

color(3,C3,O3),

color(3,C4,O4),

diferents([N1,M1,O1]),

diferents([N2,M2,O2]),

diferents([N3,M3,O3]),

diferents([N4,M4,O4]),

agafa([D1,D2,D3,D4]),

color(4,D1,P1),

color(4,D2,P2),

color(4,D3,P3),

color(4,D4,P4),

diferents([N1,M1,O1,P1]),

diferents([N2,M2,O2,P2]),

diferents([N3,M3,O3,P3]),

diferents([N4,M4,O4,P4]),

nl,

write([N1,N2,N3,N4]),nl,

write([M1,M2,M3,M4]),nl,

write([O1,O2,O3,O4]),nl,

write([P1,P2,P3,P4]).