

prov PROLOG del 17/01/2014 B

ESERCIZIO 1 [12 punti]

Scrivere un predicato `go/0` che utilizzando la libreria del CLP stampi una soluzione, se esiste, per il problema delle N regine, con N da 1 a 15, come sotto riportato.

```
?- go.  
con 1 regine[1]  
con 2 regine no soluzioni  
con 3 regine no soluzioni  
con 4 regine[2,4,1,3]  
con 5 regine[1,3,5,2,4]  
con 6 regine[2,4,6,1,3,5]  
con 7 regine[1,3,5,7,2,4,6]  
con 8 regine[1,5,8,6,3,7,2,4]  
con 9 regine[1,3,6,8,2,4,9,7,5]  
con 10 regine[1,3,6,8,10,5,9,2,4,7]  
con 11 regine[1,3,5,7,9,11,2,4,6,8,10]  
con 12 regine[1,3,5,8,10,12,6,11,2,7,9,4]  
con 13 regine[1,3,5,2,9,12,10,13,4,6,8,11,7]  
con 14 regine[1,3,5,7,12,10,13,4,14,9,2,6,8,11]  
true
```

ESERCIZIO 1 [10 punti]

Si supponga di avere nel Data Base PROLOG un insieme di fatti a due argomenti `pezzi(CODICE,QUANTITA)`, dove `CODICE` è il codice numerico di un prodotto e `QUANTITA` è la corrispondente quantità di prodotto presente nel magazzino. Scrivere un predicato `run(TOTALE)` che legga da tastiera un codice-schema `CODS` (descritto sotto) e restituisca il totale dei pezzi disponibili corrispondenti presenti. Il codice-schema `CODS` permette di specificare gruppi di codici mediante un carattere jolly, consistente nel punto interrogativo (`?`), che sostituisce singole cifre e può essere presente un numero di volte qualsiasi. Si supponga che il Data Base sia molto grande e che il numero di caratteri jolly nel `CODS` sia sempre piccolo. Ad esempio, si suppongano presenti nel Data Base i seguenti fatti

```
pezzi(23412,2). pezzi(23414,4). pezzi(23415,3). pezzi(23418,2).  
pezzi(23512,1). pezzi(23514,2). pezzi(23518,2). pezzi(23519,8).  
pezzi(23712,4). pezzi(23714,5). pezzi(23717,5). pezzi(23519,2).  
pezzi(43412,1). pezzi(53412,2). pezzi(63412,9). pezzi(26412,1).
```

Si hanno le seguenti chiamate e risposte

```
?- run(T).  
dai il codice  
|: '23?1?' .  
T = 40.
```

```
4 ?- run(T).  
dai il codice  
|: '?3?1?' .  
T = 52.
```

ESERCIZIO 1 [8 punti]

Si scriva un predicato PROLOG `conta(L, MAX)` dove `L` è una lista di elementi interi compresi tra 0 e 9.

Considerando tutte le sequenze di due interi presenti nella lista, valutare qual è la sequenza che si ripete più e restituirlo in `MAX` (come lista).

Ogni sequenza è un termine del tipo `NUMERO1/NUMERO2/RIPETIZIONI`.

In caso più sequenze abbiano lo stesso numero massimo di ripetizioni, `MAX` sarà una lista che le contiene tutte.

Se `L = [1,2,9,5,1,5,3,3,8,4,3,0,4,2,5,4,1,5,7,1]`
`MAX = [1/5/2]`

Se `L = [1,2,9,5,1,5,3,3,8,4,1,0,4,2,5,4,1,5,7,1]`
`MAX = [4/1/2, 1/5/2]`

Se `L = [1,2,9,5,7,2,9,3,8,4,1,2,2,9,5,4,9,5,1,2]`
`MAX = [2/9/3, 9/5/3, 1/2/3]`