



Práctica 2:

CUADRADO MÁGICO

OBJETIVO

- Comprender el funcionamiento del backtracking.

PLANTEAMIENTO DEL PROBLEMA

Implementar un predicado *cuadrado/2* al que se le pasa una lista y un entero S. La lista representa una matriz cuadrada y tendrá la siguiente estructura (para el caso de 3x3):

```
[ [X1, X2, X3],  
  [Y1, Y2, Y3],  
  [Z1, Z2, Z3] ]
```

donde cada sublista representa una fila de la matriz.

El predicado deberá dar valores a las variables de manera que todas las filas sumen el valor S y además no se podrán repetir números, es decir, todas las variables deberán ser distintas entre si.

Para poder generar los valores se recomienda utilizar hechos que indiquen los posibles valores que puede tomar una variable:

```
num(1) .  
num(2) .  
.....
```

De esta forma, al preguntar por el hecho *num(X)*, X podrá ir tomando los distintos valores por backtracking.

Una vez realizado este predicado, escribir un predicado *main/0*, que pida al usuario el tamaño del cuadrado y el valor de la suma, y muestre el resultado por pantalla.

NOTA: Para realizar la comprobación de que no hallan números repetidos, puede ser útil el predicado *flatten/2*, que transforma una lista de listas en una lista plana.

NORMAS DE ESTILO

Cada predicado deberá tener un comentario que explique para que sirve. Además, se deberá indicar como se utiliza cada parámetro del predicado según la siguiente notación:

+ : El argumento debe estar instanciado antes de la llamada.

- : El argumento debe ser una variable libre antes de la llamada.

? : No hay ninguna restricción sobre el argumento. Puede estar instanciado o ser una variable.

Deberá existir un comentario al principio del programa indicando nombre de la practica, autores y grupo.

DURACIÓN DE LA PRÁCTICA:

1 Sesión