Le Carré Magique.

Le carré magique d'ordre N est un carré de nombres de dimension NxN où la somme des nombres d'une ligne, d'une colonne et d'une grande diagonale est toujours la même. Le but de ce problème est de trouver un ensemble de prédicats Prolog qui permet de générer des carrés magiques d'ordre N. Ici, on ne cherchera que les carrés magique d'ordre N qui contiennent les nombres allant de 1 à N². Dans ce cas la somme des lignes, colonnes et diagonales sera toujours égale à N*(N²+1)/2.
Remarque : il existe des ordres N, pour lesquels il n'y a pas de solution de ce type.

Ex : voici un carré magique d'ordre 3 qui contient les nombres allant de 1 à 9 et où la somme des nombres des lignes, colonnes ou diagonales est toujours égale à 15.

8	3	4
1	5	9
6	7	2

<u>Représentation des connaissances.</u>
On suppose qu'une ligne d'un carré de nombres est représentée par une liste de nombres.
Ex : la première ligne du carré précédent est représentée par la liste `[8,3,4]` .
Un carré de nombres (pas forcément magique…) est représenté par une liste de lignes.
Ex : le carré précédent est représenté par la liste `[[8,3,4],[1,5,9],[6,7,2]].`
Dans la suite, on appellera diagonale 1 du carré, la diagonale de nombres qui part du coin supérieur gauche au coin inférieur droit, ici [8,5,2] et diagonale 2, la diagonale de nombres qui part du coin supérieur droit au coin inférieur gauche, ici [4,5,6].

<u>Prédicats prédéfinis.</u>
On suppose que l'on dispose des prédicats suivants.
`longueur(L,T)` qui renvoie la taille (le nombre d'éléments) de la liste L dans la variable T.
`somme(L,S)` qui renvoie la somme des éléments de la liste L de nombres dans la variable S.
`element(X,L)` qui renvoie true si l'élémnt X appartient à L. Ce prédicat marche aussi en génération, ie, qu'il peut renvoyer dans la variable X tous les éléments de la liste L.
`concat(L1,L2,R)` qui concatène les listes **`L1`** et **`L2`** dans la liste **`R`**.
Tout autre prédicat non défini devra être réécrit.

Partie I : Vérifier qu'un carré donné est magique.

1- Ecrire le prédicat **`dimension(Carre,Dim)`** qui renvoie la dimension du carré Carre dans la variable Dim.
Ex : `dimension([[1,3,2],[4,5,7],[6,8,9]],Dim).` renvoie Dim=3.

2- Ecrire le prédicat **`element_n(N,L,X)`** qui renvoie le Nième élément de la liste L dans la variable X.

3- Ecrire le prédicat **`ligne_n(N,Carre,L)`**qui renvoie la Nième ligne du carré Carre dans la variable L.
Ex : `ligne_n(2,[[1,3,2],[4,5,7],[6,8,9]],L).` renvoie L=`[4,5,7]`.

4- Ecrire le prédicat **`colonne_n(N,Carre,C)`** qui renvoie une liste qui contient les nombres de la Nième colonne du carré Carré dans la variable C.
Ex : `colonne_n(3,[[1,3,2],[4,5,7],[6,8,9]],C).` renvoie C=`[2,7,9]`.

5- Ecrire le prédicat **`colonnes(Carre,Cs)`** qui renvoie toutes les colonnes mises sous forme de liste de nombres du carré Carre dans la variable Cs.
Ex : `colonnes([[1,3,2],[4,5,7],[6,8,9]],Cs).`
renvoie `Cs=[[1,4,6],[3,5,8],[2,7,9]].`

Pour y arriver, on pourra par exemple écrire un prédicat **`colonnes(I,Carre,Cs)`** qui renvoie les colonnes `I` à `Dim` dans la variable `Cs, où Dim` est la dimension de `Carre.`

6- Ecrire les prédicats **`diagonale1(Carre,D1)`** et **`diagonale2(Carre,D2)`** qui renvoient respectivement les diagonales 1 et 2 du carré Carre dans les variables D1 et D2.
Ex : `diagonale1([[1,3,2],[4,5,7],[6,8,9]],D1).` renvoie `D1=[1,5,9]`
et `diagonale2([[1,3,2],[4,5,7],[6,8,9]],D2).` renvoie `D2=[2,5,6]`.

Pour y arriver, on pourra par exemple écrire les deux prédicats suivants :

- `diagonale_1(I,[L0,L1,…,LX],D)` qui renvoie respectivement les I^{ème}, (I+1)^{ème}, …, (I+X)^{ème} éléments des listes L0, L1, …, LX dans la variable D.
  Ex : `diagonale_1(2,[[4,5,7],[6,8,9]],D)` renvoie `D=[5,9]` et
  `diagonale_1(1,[[1,3,2],[4,5,7],[6,8,9]],D)` renvoie `D=[1,5,9].`

- `diagonale_2(J,[L0,L1,…,LX],D)` qui renvoie respectivement les J^{ème}, (J-1)^{ème}, …, (I-X)^{ème} éléments des listes L0, L1, …, LX dans la variable D.
  Ex : `diagonale_2(3,[[4,5,7],[6,8,9]],D)` renvoie `D=[7,8].`

7- Ecrire le prédicat **`composantes(Carre,Comp)`** qui renvoie les lignes, colonnes et diagonales (sous forme de liste de nombres) du carré Carre dans la variable liste Comp.

Ex : `composantes([[1,2],[3,4]],Comp).` renvoie
Comp=`[[1,2],[3,4],[1,3],[2,4],[1,4],[2,3]].`
8- Ecrire le prédicat **`magique(Carre)`** qui renvoie True si le carre Carre est magique.
PS : un carré est magiques si la somme des nombres de chaque composante est toujours la même…

Partie II : Génération d'un carré magique d'ordre N.

1- Ecrire le prédicat **`genere_liste(UB,L)`** qui génère la liste des nombres allant de 1 à UB dans la variable L. L'ordre de ces nombres n'aura aucune importance.

Ex : `genere_liste(6,L).` renvoie par ex `L=[1,2,3,4,5,6]` ou `L=[6,5,4,3,2,1]`.

2- Ecrire le prédicat **`retire_el(Liste,X,NewListe)`** qui retire la première occurrence de l'élément X trouvée dans la liste L et qui renvoie le résultat dans la variable NewListe. Si l'élément X n'appartient pas à L, alors NewListe=L.

Ex : `retire_el(=[1,2,3,4],3,L).` renvoie `L=[1,2,4]`.

3- Ecrire le prédicat **`genere_ligne(N,ListeNbs,L,NewListeNbs)`** capable de générer une liste de nombres de taille N à partir de la liste ListeNbs en renvoyant le résultat dans la variable L et la liste des nombres non utilisés dans NewListeNbs. Ce prédicat devra être capable de générer toutes les listes possibles si on demande plusieurs réponses.

Ex : `genere_ligne(2,[1,2,3,4],L,NewListeNbs).` pourra renvoyer toutes les réponses suivantes :
L = [1,2] NewListeNbs = [3,4]; L = [1,3] NewListeNbs = [2,4]; L = [1,4] NewListeNbs = [2,3];
L = [2,1] NewListeNbs = [3,4]; L = [2,3] NewListeNbs = [1,4]; L = [2,4] NewListeNbs = [1,3];
L = [3,1] NewListeNbs = [2,4]; L = [3,2] NewListeNbs = [1,4]; L = [3,4] NewListeNbs = [1,2];
L = [4,1] NewListeNbs = [2,3]; L = [4,2] NewListeNbs = [1,3]; L = [4,3] NewListeNbs = [1,2];

4- Ecrire le prédicat **`genere_carre(N,Carre)`** qui génère un carré d'ordre N (pas forcément magique) dans la variable Carre. Ce prédicat devra être capable de générer tous les carrés d'ordre N possibles…

PS : Générer un carré d'ordre N revient à générer N lignes en les mettant dans une liste.

5- Ecrire le prédicat **`solve(N,Carre)`** qui renvoie un carré magique (si il en existe 1) d'ordre N dans la variable Carre. Ce prédicat devra être capable de générer tous les carrés magiques possibles d'ordre N.