Test de magicité [tb04] - Exercice résolu

Karine Zampieri, Stéphane Rivière

Unisciel algoprog Version 19 mai 2018

Table des matières

C - Test de magicité (TP)

Mots-Clés Tableau multidimensionnel, Carré magique ■
Requis Structures de base, Structures conditionnelles, Algorithmes paramétrés, Structures répétitives, Schéma itératif, Tableaux ■
Difficulté • • ◦ (45 min) ■

Objectif
Cet exercice teste la « magicité » d'un carré d'ordre n.

1 Énoncé

Définitions

Un **carré d'ordre** n est un tableau bidimensionnel $n \times n$ d'entiers.

Il est dit **semi-magique** si la somme de chaque ligne, la somme de chaque colonne et la somme de chaque diagonale principale (montante et descendante) est la même valeur.

Enfin, il est dit **magique**[1] s'il est semi-magique et s'il est une permutation des entiers $[1..n^2]$. Dans ce cas, la somme théorique vaut $n(n^2+1)/2$.

Exemple

Le carré `[4,5,6, 1,2,3, 7,8,9]` d'ordre 3 est une 9-permutation mais n'est pas semi-magique.

4	5	6	=15
1	2	3	=6
7	8	9	=24

Exemple

Le carré `[8,1,6, 3,5,7, 4,9,2]` d'ordre 3 est magique (puisque 9-permutation et semi-magique).

			=15(diag)
8	1	6	=15
3	5	7	=15
4	9	2	=15
=15	=15	=15	=15(diag)

Exemple

Le carré suivant d'ordre 4 est un carré magique.

				=34(diag)
16	3	2	13	=34
5	10	11	8	=34
9	6	7	12	=34
4	15	14	1	=34
=34	=34	=34	=34	=34(diag)

Objectif

Représenter un carré sous sa forme linéaire, saisir un carré et tester sa magicité.

---

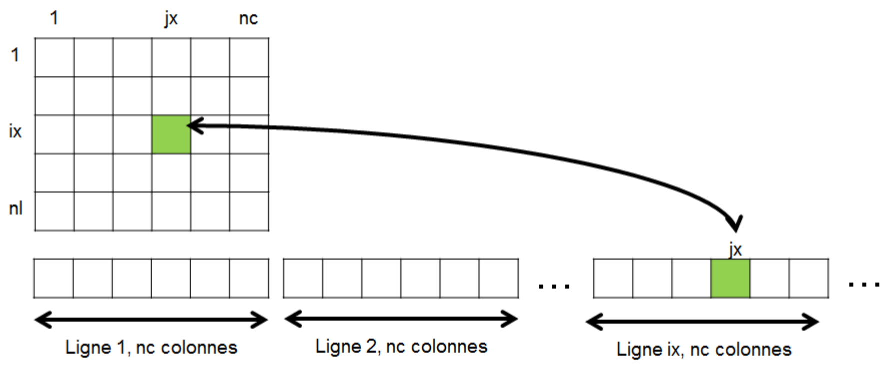
1. Les carrés magiques sont très anciens, puisqu'on trouve leur trace, il y a plus de 3000 ans, sur la carapace d'une tortue chinoise de Lo Shu. En Europe, le premier carré magique apparaît en 1514 sur une gravure du peintre allemand A. Dürer. Si durant de nombreux siècles ces carrés étaient attachés à des superstitions divines, à partir du XVIIe siècle, ils ont fait l'objet de nombreuses études mathématiques.

2 Algorithmique, Programmation

Définissez la constante `TMAX=100` (nombre maximal de cases d'un tableau), le type `ITableau` comme étant un tableau de `TMAX` entiers, puis le typ `TCarre` comme étant un `ITableau`.

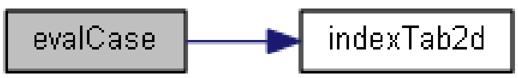
2.1 Stockage linéaire (tableau bidimensionnel)

Stockage linéaire d'un tableau bidimensionnel
Il permet d'optimiser son espace mémoire et correspond à effectuer la transformation suivante :

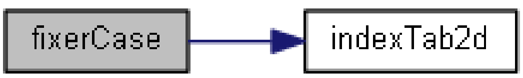


Écrivez une fonction `indexTab2d(j,k,ncols)` qui renvoie l'index linéaire de la case en `(j,k)` pour un `ITableau` à `ncols` colonnes. **Attention,** les indices sont numérotés à partir de 1.

Écrivez une fonction `evalCase(t,j,k,n)` qui renvoie la valeur de l'élément en `(j,k)` d'un `ITableau t` de `n` colonnes.



Écrivez une procédure `fixerCase(t,j,k,n,valeur)` qui fixe l'élément en `(j,k)` d'un `ITableau t` de `n` colonnes à la valeur `valeur`.



2.2 Procédure afficherCarre (affichage d'un carré)

Écrivez une procédure `afficherCarre(t,n)` qui affiche un `ITableau t` d'ordre `n` sous sa forme bidimensionnelle. Exemple d'un carré d'ordre 4 :

16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

2.3 Fonction permutationTab (test de permutation)

Définition
Une n-**permutation** est l'ensemble des entiers $\{1, 2, \ldots, n\}$.

Écrivez le **profil** d'une fonction `permutationTab(t,n)` qui renvoie `Vrai` si les `n` premières valeurs d'un `ITableau t` est une n-permutation, `Faux` sinon.

Écrivez le corps de la fonction.

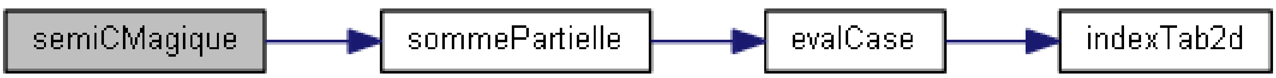
2.4 Test de magicité

Définissez le type `TCarre` comme étant un `ITableau`.

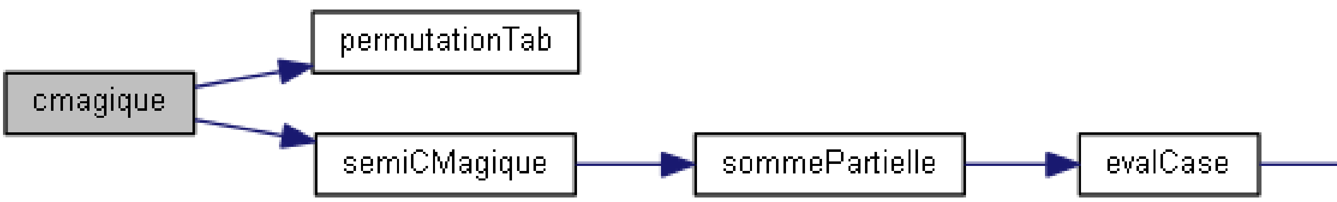
Écrivez une fonction `sommePartielle(t,ideb,jdeb,iincr,jincr,n)` qui renvoie la somme de `n` cases d'un `TCarre t` en partant de celle en `(ideb,jdeb)` et d'incrémentations de ligne `iincr` et de colonne `jincr`.

```
.        jdeb
ideb  .   .   .   .
      .   .   .   .
      .   .   .   .
      .   .   .   .
```

Écrivez ensuite une fonction `semiCMagique(t,n)` qui teste et renvoie `Vrai` si un `TCarre t` d'ordre `n` est semi-magique, `Faux` sinon.



Déduisez une fonction `cmagique(t,n)` qui teste et renvoie `Vrai` si un `TCarre t` d'ordre `n` est magique, `Faux` sinon.



2.5 Algorithme principal

Téléchargez le fichier suivant et mettez-le dans votre dossier.
C @[UtilsTB.c]

Copiez/collez ensuite les lignes suivantes :
C **Au début** de votre programme :

```
#include "UtilsTB.c"
```

Soit la fonction `saisirTab(t)` qui effectue la saisie contrôlée du nombre de valeurs (entier compris entre 1 et `TMAX`), saisit les valeurs entières dans un `ITableau t` puis renvoie l'entier du nombre de valeurs saisies.

C @[saisirTab] (dans UtilsTB)

Écrivez un programme qui :

- Déclare un `TCarre` et saisit les entiers dans le `TCarre` (fonction `saisirTab`).
- Demande et saisit l'ordre du `TCarre`.
- Enfin affiche le `TCarre` ainsi que ses caractéristiques (permutation, semi-magique, magique).

Testez avec les exemples ci-dessous :

- Le carré `[4,5,6,1,2,3,7,8,9]` d'ordre 3.
- Le carré `[8,1,6,3,5,7,4,9,2]` d'ordre 3.
- Le carré `[16,3,2,13,5,10,11,8,9,6,7,12,4,15,14,1]` d'ordre 4.