

Les tableaux à 2 dimensions : matrices

Un tableau à 2 dimensions comporte cette fois des lignes et des colonnes. Spontanément, on pense à la résolution de problèmes sur les matrices. C'est effectivement l'outil à utiliser dans ce domaine.

Un tableau à 2 dimensions se manipule comme un tableau à 1 dimension mais il faut 2 indices pour accéder à une cellule du tableau.

Autre particularité : pour parcourir un tableau à 1 dimension, une simple boucle suffit. Pour parcourir un tableau à 2 dimensions, il est nécessaire d'avoir 2 boucles imbriquées (une pour les lignes et une pour les colonnes).

Exemple de matrice

On va commencer par le calcul le plus simple sur les matrices : l'addition de 2 matrices carrées. Le but n'est pas de saisir à chaque fois les 2 éléments correspondants de chaque matrice et d'afficher immédiatement l'élément somme, sinon on n'aurait pas besoin de mémoriser les informations dans une matrice. Le but est de saisir complètement la première matrice, puis la seconde et, au final, d'afficher la matrice somme.

Somme de 2 matrices carrées :

Le résultat de la somme de 2 matrices carrées $N \times N$ est la matrice carrée $N \times N$ donc chaque élément est la somme des 2 éléments qui sont à la même position dans les 2 premières matrices..

En version algorithmique, on ne va pas s'inquiéter des positionnements pour la saisie et l'affichage. Quand on passera à la transcription codée, il faudra en tenir compte et vous découvrirez un nouvel outil.

Voici l'algorithme, plutôt simple, de la saisie de 2 matrices 4×4 et de l'affichage de la somme des 2 matrices. Pour simplifier le problème (et la transcription en C++), on ne manipulera que des entiers et, lors des tests, vous ne saisirez que des petites valeurs.

```
programme sommeMatrice
  m1[0..3, 0..3], m2[0..3, 0..3] : entier
  i, j : entier

debut
// saisie de la première matrice
  pour i de 0 à 3
    pour j e 0 à 3
      saisir m1[i, j]
    finpour
  finpour

// saisie de la seconde matrice
  pour i de 0 à 3
    pour j e 0 à 3
      saisir m2[i, j]
    finpour
  finpour

// calcul et affichage de la matrice somme
  pour i de 0 à 3
    pour j e 0 à 3
      afficher (m1[i, j] + m2[i, j])
    finpour
  finpour
fin
```

Prenez le temps de bien observer cet algorithme. Regardez comment les tableaux sont déclarés, avec 2 dimensions, et comment ils sont utilisés, avec 2 indices. Regardez bien aussi le principe des doubles boucles pour parcourir les lignes et les colonnes.

Exercice :

Coder cet algorithme