

Exercices sur les tableaux

I) Manipulations dans un tableau

On définit un tableau de taille 10, par exemple, contenant des entiers aléatoires entre 0 et 100.

- 1) Ecrire une fonction saisie fournissant un tel tableau.
- 2) Ecrire une fonction moyenne retournant la moyenne arithmétique des éléments du tableau.
- 3) Ecrire une fonction bornes retournant, lors d'une recherche simultanée, le plus petit et le plus grand des éléments.
- 4) Ecrire une fonction permuter effectuant une permutation circulaire de ses valeurs : chaque élément doit prendre la valeur du suivant, sauf le dernier qui doit prendre la valeur du premier.
- 5) Ecrire un programme qui, après avoir réalisée la procédure saisie, propose un menu permettant d'effectuer les modules 2) 3) ou 4) sur le tableau.

II) Tri par bulle

Principe : Le principe consiste à parcourir le tableau en comparant 2 à 2 des éléments voisins et en les permutant éventuellement. Après un balayage complet, le plus grand élément se trouve à l'indice n . On recommence l'opération sur les $n - 1$ éléments restants. On s'arrête lorsque la partie du tableau restant à trier est d'un seul élément ou lorsqu'un balayage complet du tableau n'a pas provoqué de permutation

Par exemple, pour trier le tableau	3 5 2 1 8 9 7 3
on obtiendra successivement	3 2 1 5 8 7 3 9 5 et 9 "remontent"
	2 1 3 5 7 3 8 9 3 et 8 "remontent"
	1 2 3 5 3 7 8 9 2 et 7 "remontent"
	1 2 3 3 5 7 8 9 5 "remonte"

Ecrire en Scilab une fonction `tri_bulle(t)`

Le nom "par bulle" vient du fait que la plus grande valeur remonte petit à petit vers le haut comme une bulle. Cette méthode est efficace pour un tableau presque trié.

III) Carré magique

On considère un tableau carré de n lignes et n colonnes. Un tel tableau est appelé **carré magique** lorsque les sommes des éléments sur une même ligne, une même colonne ou une même diagonale (principale) sont égales.

Test d'un tableau

- 1) Ecrire la fonction `[t]=Saisie_Carre` demandant la construction d'un tableau à l'utilisateur.
- 2) Ecrire la fonction `[s]=Ligne(n,t)` renvoyant la somme des éléments de la ligne n du carré t .
- 3) Ecrire de même une fonction `Colonne` et une fonction `Diagonale` (cette dernière doit avoir un paramètre supplémentaire pour déterminer laquelle des deux diagonales est concernée).
- 4) Ecrire une fonction booléenne `Magique` retournant *true* ou *false* selon que le carré t reçu en paramètre est magique ou non.
- 5) Ecrire un programme utilisant les modules précédents.

Création d'un carré magique

Voici un algorithme permettant la construction d'un carré magique d'ordre impair $n = 2p + 1$ (où le nombre d'éléments par côté est impair) et dans lequel les nombres entiers vont de 1 à n^2 .

- L'élément juste au-dessous du centre est occupé par le chiffre 1 ($t(p + 2, p + 1) := 1$).
- Les éléments suivants : 2, 3, 4, ... n^2 sont placés dans les cases se trouvant à l'intersection de la ligne du dessous et de la colonne de droite.
- Arrivé au bord du carré, on poursuit l'opération à l'extrémité opposée en suivant la même règle. On identifie donc les colonnes $n+k$ et k ainsi que les lignes $n+k$ et k ($k \neq 0$)
- Si la case suivante est déjà remplie, le nombre est placé dans la même colonne, deux lignes en dessous.

Ex : 4 9 2
 3 5 7 somme = 15
 8 1 6

Ecrire un programme qui engendre puis affiche un carré magique au moyen de cet algorithme.