

Structures et matrices

1. Tableaux de points.

Pour tester vos fonctions vous pourrez utiliser le tableau désordonné des points (i, j) pour $1 \leq i, j \leq 6$ disponible dans le fichier *points.c*.

- Définissez un type *point* qui est une structure à deux champs entiers nommés x et y .
- Concevez et codez une fonction *afficher_point* qui affiche un point passé en paramètre.
- Concevez et codez une fonction *afficher_point_tab* qui affiche un tableau de points passé en paramètre avec sa taille.
- On souhaite modifier la fonction bulle pour qu'elle fonctionne avec un tableau de points. Pour cela il faut savoir comparer deux points (savoir lequel est le plus grand). On dit que l'on "dispose d'un ordre sur les points". On s'intéresse en particulier à :

l'ordre **lexicographique** : un point $p = (x_p, y_p)$ est supérieur à $q = (x_q, y_q)$ si et seulement si $x_p > x_q$ OU $(x_p = x_q$ ET $y_p > y_q)$.

l'ordre **"diagonal"** : un point $p = (x_p, y_p)$ est supérieur à $q = (x_q, y_q)$ si et seulement si $x_p + y_p > x_q + y_q$ OU $(x_p + y_p = x_q + y_q$ ET $x_p > x_q)$.

Faire un dessin pour expliquer ces deux ordres.

- Modifiez la fonction bulle pour qu'elle fonctionne avec un tableau de points et l'ordre lexicographique.
- Modifiez la fonction précédente pour qu'elle fonctionne avec un tableau de points et l'ordre diagonal.

2. Matrices.

Définir un nouveau type de donnée matrice représentant les matrices $\mathbb{N} \times \mathbb{N}$ de réels, puis concevoir et écrire les fonctions suivantes :

- affiche* qui affiche une matrice.
- nombreDeZeros* qui compte le nombre d'éléments nuls dans une matrice.
- estDiagonale* qui teste si une matrice est diagonale.
- estSymetrique* qui teste si une matrice est symétrique.
- transpose* qui remplace une matrice par sa transposée.
- addition* qui additionne deux matrices et renvoie le résultat en modifiant un argument.
- multiplication* qui effectue le produit de deux matrices et renvoie le résultat en modifiant un argument.