

## TP Informatique Pour Scientifiques

### TP 2 - Les enregistrements

---

#### Exercice 1 \_\_\_\_\_ Point 2D

Dans cet exercice on va définir le type enregistrement `point2D` et apprendre à le manipuler grâce à quelques primitives.

1. Créez un fichier `point2D.h` contenant la déclaration du type `point2D` qui comprend deux champs : `abscisse` et `ordonnee`, tous deux de type flottant double précision.
2. Écrivez les opérations primitives pour le type `point2D` à savoir les constructeurs et les opérations d'accès.
3. Créez le programme `test_point2D.c` pour tester le type et les opérations définies ci-dessus. Rappel : l'en-tête de votre programme doit contenir (juste après les lignes de commentaires indiquant votre nom et la date) la commande `#include "point2D.h"`
4. Implantez les opérations non primitives suivantes :
  - (a) `void afficher_point2D (point2D P)` ; qui affiche l'abscisse et l'ordonnée d'un point passé en paramètre,
  - (b) `double distance (point2D P, point2D N)` ; qui calcule la distance euclidienne entre deux points (attention, la racine carrée se calcule par la commande `sqrt` incluse dans la bibliothèque `math`. Il faut donc ajouter dans l'en-tête de votre fichier l'instruction `#include <math.h>` et compiler avec l'option `-lm` pour que le compilateur trouve la bibliothèque).
  - (c) `point2D milieu (point2D P, point2D N)` ; qui prend en paramètres deux points et renvoie un point qui se situe au milieu des deux autres.
5. Ajoutez au fichier `test_point2D.c` les tests des opérations non primitives.

#### Exercice 2 \_\_\_\_\_ Point 3D

Répondez à nouveau à tous les points de l'exercice précédent mais cette fois-ci avec un type `point3D` décrivant les coordonnées d'un point dans un espace à trois dimensions (il a donc un champ supplémentaire appelé `cote`).

#### Exercice 3 \_\_\_\_\_ Transformations géométriques

1. Créez une nouvelle opération `point2D symetrique2D (point2D P, point2D C)` ; qui prend en paramètres deux points et renvoie un point qui est l'image de `P` par une symétrie de centre `C`.
2. Même chose pour l'opération `point2D homothetie2D (point2D P, point2D C, double r)` ; qui donne l'image du point `P` par une homothétie de centre `C` et de rapport `r`.
3. Même chose pour l'opération `point2D rotation2D (point2D P, double a)` ; qui calcule l'image d'un point `P` par une rotation de centre l'origine et d'angle `a` (exprimé en radians).
4. Que faudrait-il changer pour écrire une fonction réalisant une rotation de centre un `point2D` quelconque ?