Travaux pratiques n° 6 : "Géométrie" dans le plan et ensembles

Exercice 1

1. Réaliser une classe `point` permettant de manipuler un point dans le plan.
2. Écrire une fonction `Init_point`, de type de retour `void`, prenant parmi ses argument entre autres deux flottants et qui initialise les coordonnées d'un point avec les flottants donnés en argument.
3. Écrire une fonction `Deplace`, de type de retour `void`, effectuant une translation définie par ses deux arguments (`float`).
4. Écrire une fonciton `Affiche` qui se contente d'afficher les coordonnées cartésiennes du point.
5. Écrire une petit programme d'essai (`main`) déclarant un point, l'affichant, le déplaçant et l'affichant de nouveau.

Exercice 2

Ajouter au programme précédent de nouvelles fonctions (toutes de type de retour `void`) :
– `Homothetie` qui effectue une homothétie dont le rapport est fourni en argument.
– `Rotation` qui effectue une rotation dont l'angle est fourni en argument.
– `Rho` et `Theta` qui fournissent en retour les **coordonnées polaires d'un point** (on choisit une détermination de l'angle θ entre $-\pi$ et $+\pi$, avec un angle valant 0 si l'abscisse du point est nulle).

Exercice 3

Modifier le programme précédent de façon à ce qu'un point soit représenté par ses coordonnées polaires et non plus cartésiennes. Cependant les fonctions `Init_point` et `Deplace` reçoive toujours des coordonnées cartésiennes en entrée, et `Affiche` affiche les coordonnées cartésiennes d'un point. (Puisqu'on travaille en coordonnées polaires nous supposons que le point considéré est distinct du point de coordonnées cartésiennes $(0, 0)$.)

Exercice 4

Réaliser une structure `set_char` permettant de manipuler des ensembles de caractères. On devra pouvoir réaliser les opérations classiques suvantes : lui ajouter un élément, connaître son cardinal, savoir si un caractère donné lui appartient.

Ici on n'effectuera aucune allocation dynamique d'emplacements mémoire. Il faudra donc prévoir, dans la structure, un tableau de taille fixe.