

**Algorithmique et structures de données 1**

## TP 5

**Exercice 1 :** Ecrire un programme C qui définit une structure *point* qui contiendra les deux coordonnées d'un point du plan. Puis lit deux points et affiche la distance entre ces deux derniers. La distance entre deux points  $(x_1, y_1)$  et  $(x_2, y_2)$  est :

$$\text{Distance} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

**Exercice 2 :** Ecrire un programme C qui définit une structure *etudiant* où un étudiant est représenté par son nom, son prénom et une note. Lit ensuite une liste d'étudiants entrée par l'utilisateur et affiche les noms de tous les étudiants ayant une note supérieure ou égale à 10 sur 20.

**Exercice 3 :** Écrire un programme en C pour créer et stocker des informations dans un fichier texte (test.txt). Par exemple :

- Données de test : Entrez le contenu à stocker dans le fichier : Ceci est le contenu du fichier.
- Résultat attendu : Le fichier créé et enregistré avec succès...

**Exercice 4 :** Écrire un programme C pour lire un fichier et afficher son contenu sur la console.

**Exercice 5 :** Écrire un programme C qui permet de saisir n'importe quel nombre de l'utilisateur et trouver le cube du nombre donné en utilisant une fonction.

**Exercice 6 :** Écrire un programme en C qui permet de saisir le rayon d'un cercle et trouver le diamètre, la circonférence et l'aire du cercle donné en utilisant les fonctions.

**Exercice 7 :** Écrire un programme en C qui permet de vérifier si un nombre est premier, parfait ou non en utilisant des fonctions. Un nombre entier est dit parfait s'il est égal à la somme de ses diviseurs propres (diviseurs de ce nombre autres que lui-même). Par exemple, le nombre 6 est un nombre parfait. En effet 1, 2 et 3 sont les diviseurs propres de 6 et  $1+2+3=6$ . 28 est également un nombre parfait :  $1+2+4+7+14=28$ .

**Exercice 8 :** Écrire un programme en C qui permet d'obtenir le plus grand élément d'un tableau en utilisant la fonction.