

## TD7 – Structures de données

---

### Objectifs :

- Utiliser une structure de données existante
  - Utiliser les fonctions/procédures qui permettent de manipuler un TAD
  - Créer une nouvelle structure de données
  - Créer les fonctions/procédures qui permettent de manipuler un nouveau TAD
- 

On considère la structure de données Point2D présentée en cours :

```
typedef struct {  
    int x ;  
    int y ;  
} Point2D ;
```

Récupérez les fonctions et les procédures permettant de manipuler le type Point2D qui se trouvent dans le fichier TADPoint2D.c et incluez ce fichier dans votre projet. Créez le fichier TADPoint2D.h qui contient les prototypes correspondants.

### Exercice 1 :

Écrivez un programme principal qui :

- Déclare deux points P1 et P2,
- Crée le point P1 avec les valeurs 2 et 4
- Crée le point P2 comme une copie du point P1
- Affiche le point P1
- Modifie le point P2 de la manière suivante :
  - Nouvelle abscisse = le double de l'abscisse de P2
  - Nouvelle ordonnées = le triple de l'ordonnée de P2
- Affiche le point P2

**Exercice 2 :**

Écrire une procédure **afficheFigure** qui affiche un tableau de Point2D. La taille effective du tableau doit faire partie des paramètres. L’affichage d’un tableau de Point2D doit bien évidemment utiliser la procédure qui affiche un Point2D.

**Exercice 3 :**

Écrire une procédure **initFigure** qui a comme paramètres un tableau de Point2D et une taille et qui initialise les points du tableau en utilisant la fonction **saisieEntier** que nous avons déjà utilisé dans les TD précédents (elle se trouve dans le fichier TADPoint2D.c) ou la fonction **nombreAleatoire** suivante :

```
int nombreAleatoire (int valMax)
{
    return (rand ()%(valMax+1)) ;
}
```

**Exercice 4 :**

Écrire une fonction **distance** qui retourne la distance entre 2 Point2D.

Rappel : distance entre P1 et P2 =  $\sqrt{((x2 - x1)^2 + (y2 - y1)^2)}$

**Exercice 5 :**

Écrire une fonction **perimetreFigure** qui calcule le périmètre d’un tableau de Point2D. La taille du tableau fait également partie des paramètres de la fonction.

On considère que la figure est fermée, c’est-à-dire qu’il faut également compter dans le périmètre la distance entre le dernier point du tableau et le premier.

**Exercice 6 :**

Créez un programme principal qui :

- Déclare un tableau de Point2D de taille MAXTAB nommé Figure
- Initialise le tableau Figure à l’aide de la procédure **initFigure**
- Affiche le tableau Figure
- Calcule et affiche le périmètre du tableau Figure

**Exercice 7 :** Les planètes d’un système planétaire

Les exercices suivants vont vous permettre d’écrire une application relative au traitement des données d’un système planétaire. On caractérise chaque planète d’un système planétaire par :

- son nom (20 caractères au maximum),

- sa densité (float),
- la distance moyenne de l'étoile autour de laquelle elle gravite (float) en 10E6 km,
- son nombre de satellites (int).

1) Définir un type structuré permettant de représenter une planète ainsi que son TAD complet. Attention quand vous manipulez le champ nom de la structure : l'affectation ou l'égalité ne fonctionnent pas avec des tableaux de caractères : il faut utiliser des fonctions de la bibliothèque string.h –strcmp, strcpy, ...).

2) Écrire une procédure afficherPlanete(...), qui affiche clairement les caractéristiques de la planète reçue en paramètre.

3) Écrire une fonction int egales(...) qui compare les caractéristiques des planètes p1 et p2 passées en paramètre de la fonction. La fonction retourne 0 si les planètes ont des caractéristiques différentes, et 1 si les deux planètes ont les mêmes caractéristiques.

4) Écrire une fonction recherchePlanète qui recherche, dans un système planétaire comportant n planètes, la présence d'une planète particulière passée en paramètre. La fonction doit retourner la position de la planète dans le tableau représentant le système planétaire, ou -1 si la planète n'a pas été trouvée.

5) Écrire une procédure qui calcule le nombre moyen de satellites par planète ainsi que la densité moyenne des planètes d'un système de n planètes