

# **Les Structures**

## **Exercice 1**

Définition de nom de type

- ❖ Définir un type Date pour des variables formées d'un numéro de jour, d'un nom de mois et d'un numéro d'année.
- ❖ Ecrire des fonctions de lecture et d'écriture d'une variable de type Date.
- ❖ Dans un premier temps, on ne se préoccupera pas de la validité de la date entrée.

## **Exercice 2**

Ecrire la déclaration d'un type Fiche permettant de mémoriser les informations sur un étudiant :

- ✓ son nom ;
  - ✓ son prénom ;
  - ✓ sa date de Naissance, de type Date ;
  - ✓ sa formation, représentée par cinq lettres ;
  - ✓ s'il est redoublant ou non ;
  - ✓ son groupe de TD, représenté par un entier ;
  - ✓ ses notes, représentées par un tableau note d'au plus MAXNOTES réels;
  - ✓ un entier nbnotes indiquant le nombre de notes valides dans le tableau note.
- ❖ Ecrire les fonctions LireFiche et EcrireFiche de lecture et d'écriture (I/O) d'une Fiche. Aucune note n'est entrée par la fonction LireFiche.
  - ❖ Ecrire une fonction AjouteNote qui reçoit une Fiche et ajoute une note, si cela est possible.
  - ❖ Ecrire une fonction Moyenne qui reçoit une Fiche et renvoie, si cela est possible, la moyenne des notes de l'étudiant.

### **Exercice 3 :**

Donner une fonction qui retourne la moyenne, le minimum et le maximum d'un tableau unidimensionnels ;

### **Exercice 4 :**

Soit la structure suivante :

```
Struct s_point{  
    Char c ;  
    Int x,y;  
};
```

Écrire une fonction qui reçoit en argument une structure de type s\_point et qui affiche le contenu sous la forme :

point B de coordonnées 10 20

- a) En transmettant en argument la valeur de la structure concernée
- b) En transmettant en argument l'adresse de la structure concernée

Dans les deux cas, en écrira un petit programme d'essai de la fonction ainsi réalisée.