

E1**Partie I – structure Date**

Soit une structure pour représenter des dates :

```
struct Date
{
    int jour;
    int mois;
    int annee;
};
```

Définir une constante ANNEE_DE_BASE que l'on mettra à la valeur que l'on souhaite.

En passant les structures en paramètre par référence, écrire :

- a) une fonction qui initialise une Date avec trois paramètres entiers, l'année ayant comme valeur par défaut l'année de base,
- b) une fonction qui affiche une Date,
- c) une fonction qui retourne un booléen indiquant si deux variables Date sont identiques
- d) une fonction qui surcharge la précédente, prenant en paramètre une Date et trois entiers,

Tester les fonctions dans un programme principal.

Partie II – structure ListeDate

Soit une structure pour représenter un ensemble de dates (sous forme de tableau) :

```
struct ListeDates
{
    int nbdates;
    Date* dat;
};
```

En passant les structures en paramètre par référence écrire :

- a) une fonction qui initialise une ListeDates avec un paramètre entier indiquant le nombre de dates dans la liste ; cette fonction devra allouer dynamiquement le tableau de Date et l'initialiser à tous les premier janvier à partir de l'année de base,
- b) une fonction qui affiche une ListeDates,
- c) une fonction qui modifie la Date positionnée à un rang donné dans une ListeDates,
- d) une fonction qui libère la place mémoire réservée dynamiquement pour une ListeDates

Tester les fonctions en complétant le programme principal.