

**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 :

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

Tester les fonctions dans un programme principal.

*integer*

**Partie II – structure ListeDate**

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

```
struct ListDates
{
    int nbjours;
    Date* dat;
};
```

En passant les structures en paramètre par référence écrire :

- une fonction qui initialise une ListDates avec un paramètre entier indiquant le nombre de jours 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, *动态分配, 用指针*
- une fonction qui affiche une ListDates, *listDate.dat = new Date[c];*
- une fonction qui modifie la Date positionnée à un rang donné dans une ListDates,
- une fonction qui libère la place mémoire réservée dynamiquement pour une ListDates

*释放内存*

*delete[] ld.date;*

Tester les fonctions en complétant le programme principal.