

# Fonctions, structures et tableaux des structures

Chaque fonction doit être testé dans un main()

Aucune fonction ne doit comporter un cin ou un cout (sauf les fonction de saisie et d'affichage)

La structure date est définie comme suit :

```
struct date
{
    int j ;
    int m ;
    int a ;
} ;
```

1- Ecrire une fonction egale :

```
bool egale(date & d1, date &d2) ;
```

qui retourne vrai si les deux dates d1 et d2 sont identiques.

2- Ecrire une fonction qui prend en argument un tableau T de dates, sa taille maximale Nmax et qui demande à l'utilisateur de saisir le nombre de dates N qu'il veut saisir dans le tableau (faites en sorte que  $N < N_{\max}$ ). Ensuite la fonction doit retourner la valeur de N.

3- Ecrire une fonction d'affichage :

```
void afficher(date T[], int N) ;
```

4- Ecrire une fonction qui prend en argument un tableau T , sa taille N et une date D. La fonction doit envoyer vrai si D appartient au tableau.

5- Ecrire une fonction *anterieure* qui prend en argument 2 dates d1 et d2 et qui renvoie vrai si d1 est antérieure à d2.

6- Ecrire une fonction qui insère une date D dans un tableau de N dates supposé trié. Après insertion le tableau doit rester trié. Voici la stratégie :

- Insérer D à la case d'indice N dans le tableau
- Initialiser i à N
- Tant que  $i > 0$  et tant que  $T[i-1] \geq T[i]$  faire :
  - Échanger  $T[i]$  et  $T[i-1]$
  - $i = i - 1$

7- **Facultative** : utiliser la fonction de la question 6 pour trier un tableau de dates.

Voici la stratégie :

- Pour i allant de 1 à N-1 faire
  - appeler la fonction de la question 6 ( à quelques modifications près)

Un compte client est caractérisée par un nom, un prénom, une date de naissance et un solde.

8- Écrire la structure « compte »

9- Ecrire une fonction qui prend en argument un tableau de comptes et qui retourne le solde minimal et le solde maximal et la moyenne des soldes.