

**Fonctions**  
**Travaux Dirigés – Séance n. 4**

---

## 1 Structuration du programme « date du lendemain »

Nous allons reprendre le programme de la date du lendemain et le construire progressivement à l'aide de fonctions. Vous allez voir comment les fonctions vous permettront de structurer de façon cohérente votre programme.

**exercice 1)** Écrivez la fonction `ecrireDate` qui prend 3 paramètres, jour, mois, et année et qui écrit sur la sortie standard la date avec le mois en toutes lettres. Cette fonction possède l'en-tête suivant :

```
/* Antécédent: jour, mois, annee représentent une date valide
   Conséquent: la date est écrite sur la sortie standard
               avec le mois en toutes lettres
*/
void écrireDate(int jour, int mois, int annee)
```

**exercice 2)** Écrivez la fonction `main` qui lit sur l'entrée standard trois entiers : jour, mois et année et qui écrit sur la sortie standard la date à l'aide de la fonction précédente `ecrireDate`.

**exercice 3)** Testez votre programme.

**exercice 4)** Écrivez maintenant la fonction `bissextile` qui teste si une année passée en paramètre est bissextile ou non. L'en-tête de la fonction est le suivant :

```
/* Antécédent: annee '(\|geqslant\|)' AnnéeMin */
   Conséquent: bissextile = vrai si l'année est bissextile, et
               faux sinon
*/
int bissextile(int annee)
```

**exercice 5)** Écrivez la fonction `joursDansMois` qui retourne le nombre de jour d'un mois d'une année donnée. Le mois et l'année sont passés en paramètre à la fonction. Vous utiliserez `bissextile` dans cette fonction. Son en-tête est le suivant :

```
/* Antécédent: mois, annee, deux entiers qui représentent un mois
               et une année valides
   Conséquent: retourne le nombre de jour dans le mois ('>= '28 et '<= '31)
*/
int joursDansMois(int mois, int annee)
```

**exercice 6)** Écrivez la fonction `valide` qui teste si une date représentée par 3 entiers (jour, mois, année) passés en paramètre est valide ou non. Vous utiliserez la fonction précédente `joursDansMois` pour écrire la fonction `valide`.

**exercice 7)** Enfin, en utilisant les fonctions précédentes, écrivez le programme du calcul de la date du lendemain en suivant l'algorithme :

```
{ lire le jour, le mois et l'année }  
{ vérifier si la date est valide }  
si non valide alors  
    { signaler l'erreur }  
sinon  
    { calculer la date du lendemain }  
    { afficher la date du lendemain }  
fin si
```

**exercice 8)** Testez votre programme.