

Types composés

Dans ce TP, vous allez programmer des types permettant de gérer des adresses.

On va considérer plusieurs types :

- le type `personne` composé d'un élément de type `etatcivil`, d'un élément de type `adresse` et d'une chaîne de caractères ;
- le type `etatcivil` composé de trois chaînes de caractère, d'un élément de type `date` et d'un booléen ;
- le type `date` composé de trois entiers ;
- le type `adresse` composé de deux entiers et de deux chaînes de caractère.

Exercice 1

Écrire une fonction qui prend en entrée le numéro d'un mois (entre 1 et 12) et un booléen indiquant si l'année est bissextile et renvoie le nombre de jours de ce mois.

Écrire une fonction qui prend en entrée un entier représentant une année et renvoie un booléen indiquant si cette année est bissextile (une année est bissextile si elle est divisible par 4 mais pas par 100 ou si elle est divisible par 400).

Construire le type `date` défini par trois entiers : le jour, le mois et l'année. Le constructeur vérifiera que les entiers forment une date valide et, sinon, renverra un message d'erreur indiquant d'où vient l'erreur.

Écrire une fonction `string_of_date` qui, à partir d'un élément de type `date` rend une chaîne de caractère contenant la date (vous utiliserez pour cela l'opérateur `^` de concaténation de chaînes).

Exercice 2

Construire le type `etatcivil` composé de :

- une chaîne de caractères pour le prénom ;
- une chaîne de caractères pour le nom de famille ;
- un élément de type `date` pour la date de naissance ;
- une chaîne de caractères pour la ville de naissance.

Écrire une fonction `string_of_etatcivil` qui affiche l'état civil sous la forme `<prenom> <nom>, ne le <date> a <ville>`

Exercice 3

Construire le type `adresse` composé de :

- un entier pour le numéro dans la rue ;
- une chaîne de caractères pour le nom de la rue ;
- un entier pour le code postal ;
- une chaîne de caractères pour la ville.

Écrire une fonction `string_of_adresse` qui, à partir d'un élément de type `adresse` rend une chaîne de caractères de la forme : `<numero> <rue>, <code> <ville>`

Exercice 4

Construire le type `personne` composé de

- un élément de type `etatcivil`
- un élément de type `adresse`
- une chaîne de caractères pour le numéro de téléphone.

En utilisant le caractère `'\n'` permettant d'effectuer un retour chariot, écrire une fonction `string_of_personne` qui, à partir d'un élément de type `personne` renvoie une chaîne de caractères sous la forme :

```
<etatcivil>
```

```
Residence: <adresse>
```

```
Telephone: <telephone>
```

Tester votre fonction à l'aide de la fonction `print_string`.

Exercice 5

Écrire des fonctions permettant, à partir d'un élément de type `personne`, d'extraire directement :

- le nom
- le prénom
- le numéro de la rue
- l'année de naissance