Travaux Dirigés n°7

Cours de Langage C

—Licence Informatique—

Ce TD porte sur le typage d'expression, les pointeurs de fonctions et les fichiers.

▸ **Exercice 1.** (typage d'expression) *Donner des déclarations de variables de type :*

1. *Pointeur sur pointeur sur* `char`.
2. *Tableau de* 25 *pointeurs sur* `int`.
3. *Pointeur sur tableau de* 25 `int`.
4. *Pointeur sur fonction ne prenant pas de paramètre et retournant* `void`.
5. *Tableau de* 12 *pointeurs sur fonction retournant un pointeur sur un tableau de* 7 `char`, *et prenant en paramètre un* `int`.

▸ **Exercice 2.** (utilisation d'un pointeur sur fonction) *La librairie standard met à disposition la fonction* `qsort` *qui implémente l'algorithme de tri rapide d'un tableau de données. Le prototype de cette fonction est le suivant :*

```
void qsort(void *base, size_t nel, size_t size, ... compar ...);
```

Le premier paramètre est un pointeur sur la zone à trier, le deuxième est le nombre d'éléments à trier et le troisième donne la taille de chaque élément. Le quatrième argument est la fonction de comparaison de deux éléments. Cette fonction retourne une valeur inférieure à, égale à ou supérieure à zéro, selon que son premier argument est inférieur à, égal à ou supérieur à son deuxième argument.

1. *Quel doit être le type du paramètre* `compar` *?*
2. *Donner une fonction* `comp_int` *comparant deux entiers. La fonction retournera* 0 *si les deux entiers sont égaux,* -1 *si le premier est supérieur au deuxième et* 1 *sinon.*
3. *Comment changer le type de l'expression* `comp_int` *afin qu'il devienne du même type que* `compar` *de sorte que* `comp_int` *puisse être un argument de la fonction* `qsort` *?*
4. *Donner en utilisant la fonction* `comp_int` *un exemple d'appel à* `qsort` *permettant de trier en ordre décroissant un tableau d'entiers.*

▸ **Exercice 3.** (traitement de fichiers) *On souhaite traiter un fichier ayant la structure suivante :*

- *une zone contenant le nombre n d'objets que contient le fichier ;*
- *une suite de n objets de type :*

```
typedef struct str_pers
{ char nom[20];
  char prenom[20];
  int age;
} personne;
```

1. *Donner une fonction ouvrant un fichier existant et retournant le nombre de personnes stockées dans le fichier.*
2. *Donner une fonction permettant de récupérer la i-ième personne du fichier.*
3. *Donner une fonction permettant d'ajouter une personne en fin de fichier.*