

2 Tri bulle générique

Tri Bulle. Le tri bulle consiste à parcourir la table à trier en intervertissant toute paire d'éléments consécutifs non ordonnés afin qu'après un parcours, le plus grand élément se retrouve à la fin de la table. On recommence cette opération avec la table considérée sans le dernier éléments.

Questions.

1. Donnez la définition de la fonction de prototype :

```
void TriBulle(int *,int) ;
```

qui tri — de manière destructive — un tableau d'entiers passé en premiers paramètre et dont la taille est passée en second paramètre.

2. Donnez la définition de la fonction de prototype :

```
void TriBulleGenerique (void *tab,int tabsz, int cellsize, int (*compar)(void*, void *)) ;
```

qui tri — de manière destructive — un tableau dont l'origine est passée en premiers paramètre, la taille en second paramètre, la taille (en octet) de chacune de ces cellules en troisième et une fonction de comparaison de cellule en dernier.

3 Matrices pleines de dimensions variables

La taille d'une matrice pleine de dimensions variables n'est pas connue à la compilation mais seulement lors de sa création lors de l'exécution. Les coefficients de cette matrice sont des entiers machines signés.

On se donne la description des types suivante :

- le type `matrix_t` est un pointeur sur des objets définis suivant le modèle d'identificateur `matrix_m`;
- un tel objet est :
 - soit un objet d'identificateur `zero` étant vrai si la matrice est nulle et faux dans le cas contraire,
 - soit un objet suivant le modèle d'identificateur `truematrix_m`.
- un objet de la famille d'identificateur `truematrix_m` est composé :
 - d'un entier signé d'identificateur `nblig` codant le nombre de lignes;
 - d'un entier signé d'identificateur `nbcol` codant le nombre de colonnes;
 - d'un pointeur sur des entiers signés d'identificateur `body`.

On souhaite disposer de l'opération d'addition de telles matrices : l'addition de deux matrices n'est possible que si ces matrices ont le même nombre de colonnes et le même nombre de lignes.

Une variable du type `matrix_t` vaut `NULL` si elle ne représente pas une matrice valide (par exemple, si on tente de construire une matrice avec une dimension négative ou si on tente d'additionner 2 matrices de tailles différentes).

Question.

1. Donnez les déclarations des types décrits ci-dessus.
2. Donnez la définition d'une fonction de prototype

```
matrix_t makenullmatrix ();
```

qui construit une matrice zéro;

3. Donnez la définition d'une fonction de prototype

```
matrix_t makematrix (int, int);
```

qui construit une matrice dont le nombre de lignes est passé en premier paramètre et le nombre de colonnes en second (si un de ces entiers est négatif ou nul, on retourne `NULL`). Cette fonction réserve de l'espace pour les coefficients mais n'affecte pas cet espace.