

Module 2 : Tableaux – Compléments

Objectifs

- Manipulation des tableaux triés et non triés et utilisation des blocs logiques : écrire, comprendre/corriger ou mettre à jour des diagrammes d'actions/modules

Références

- Cours de théorie OED

Partie 1

Objectif : Mise à jour des tableaux triés.

Pour tous les exercices, ne traitez les cas d'erreur que s'ils sont explicitement demandés.

Exercice 1

Un artisan conserve les caractéristiques des jouets qu'il fabrique dans le tableau **jouets** constitué de **nbJouets** cellules dont chacune concerne un jouet et reprend :

- sa référence (alphanumérique),
- son nom,
- son prix de gros,
- un tableau **matériaux** de 5 cellules maximum dont chaque cellule contient le libellé d'un matériau dont est constitué le jouet, s'il y a moins de 5 matériaux, les derniers libellés sont vides,
- le nombre d'exemplaires en stock.

Ce tableau est **trié par référence croissante**.

Cet artisan désire ajouter une série de nouveaux jouets dans le tableau.

1. Un module permettant l'obtention de chaque nouveau jouet a été écrit par un programmeur débutant. Il a pour but d'obtenir, via une fiche, la référence d'un nouveau jouet (« XXX » pour terminer), son nom, son prix de gros, le libellé de chacun des matériaux dont il est constitué (« ZZZ » pour terminer) et le nombre d'exemplaires en stock.
 - a. Effectuez la description complète de la fiche.
 - b. La signature du module et/ou son code comportent de nombreuses erreurs. Corrigez-les.



```
* fiche
Obtenir fiche.reference
ficheExiste = fiche.reference ≠ " "
  if(ficheExiste)
    Obtenir fiche.prix
    while (matériau ≠ "zzz")
      Obtenir matériau
      i++
    obtenir fiche.nbStock
  sortir fiche
```

2. Ecrivez le module ***miseAJourJouets*** qui, à l'aide des fiches obtenues une par une, met à jour le tableau **jouets** tout en le gardant trié à tout instant.

Exercice 2

L'artisan cité à l'exercice 1 désire abandonner la fabrication d'un jouet. Ecrivez le module qui reçoit la référence de ce jouet et qui le supprime du tableau. Prévoyez le cas d'une erreur dans la référence. Le module renverra un booléen indiquant si la suppression a été réellement effectuée ou non.

Exercice 3

Ce même artisan décide de supprimer un matériau qu'il utilisait pour fabriquer un certain jouet.

Ecrivez le module qui reçoit la référence du jouet et le libellé du matériau et qui supprime ce matériau du tableau **jouets**.

Version 1 : sans erreur ;

Version 2 : erreur possible dans la référence du jouet et/ou le matériau-> afficher le message d'erreur adéquat.

Exercice 4

Une association de défense des consommateurs a retenu pour un certain nombre de jeux vidéo, des informations concernant *les trois meilleures offres de prix*. On dispose ainsi du tableau **jeuxVidéo** de **nbJeux** cellules décrit ci-dessous.

jeuxVidéo	{	cellule (nbJeux *)	}	nom catégorie offres	{	cellule (3*)	}	prix nomMagasin adresseMagasin
------------------	---	-------------------------------	---	-------------------------------------	---	-----------------	---	---

Le tableau **jeuxVidéo** est trié par ordre alphabétique sur le nom du jeu et le tableau **offres** est rangé par prix croissant.

1. Afin de mettre à jour ce tableau avec les dernières statistiques du marché, vous disposez de fiches sur lesquelles apparaissent les nom, catégorie et prix d'un jeu ainsi que le nom et l'adresse du magasin dans lequel il est vendu.
Écrivez le module qui met à jour le tableau **jeuxVidéo** sachant que toutes les fiches concernent des jeux déjà présents dans le tableau mais des nouveaux magasins.
2. Un jeu présente des bugs importants. Ecrivez le module qui reçoit en paramètre le nom de ce jeu et qui le supprime du tableau. Prévoyez le cas d'une erreur dans le nom du jeu.

Partie 2

Objectif : repérer et utiliser les blocs logiques

Exercice 5

Le tableau **jouets** décrit à l'exercice 1 est à présent trié par prix de gros, du jouet le plus cher au moins cher.

1. Cela entraîne-t-il un bloc logique ? Expliquez votre réponse.
2. Et si, pour un même prix, il est trié par ordre décroissant sur le nombre d'exemplaires en stock, cela modifie-t-il la réponse précédente ?

Exercice 6

Le tableau **jeuxVidéo** de l'exercice 4 est à présent rangé par ordre alphabétique sur la catégorie (action, logique, stratégie...) et pour une même catégorie par ordre alphabétique sur le nom du jeu. Pour rappel, le tableau **offres** est trié par prix croissant.

1. Le tableau présente dès lors un niveau de bloc logique. Expliquez.

2. Ecrivez un module qui reçoit une catégorie de jeu et un nom de jeu et qui renvoie son prix le moins cher. Le module renverra -1 s'il y a une erreur dans la catégorie ou dans le nom du jeu.
3. Ecrivez le module qui affiche pour chaque catégorie de jeu, le nom et le prix du jeu le moins cher ainsi que le nom de la catégorie à laquelle il appartient.
4. Ecrivez le module qui affiche, pour l'ensemble des catégories, le nom et le prix du jeu le moins cher ainsi que le nom de la catégorie à laquelle il appartient.

Exercice 7

On dispose d'un tableau **ouvriers** de 50 cellules. Chaque cellule contient, entre autres, les heures prestées par un ouvrier chez un client au cours d'une seule semaine de 5 jours.

Chaque cellule du tableau reprend :

- le nom de l'ouvrier : **nom**,
- le nom du client chez qui il a travaillé : **nomClient**,
- pour chaque jour de la semaine, le nombre d'heures prestées chez ce client : **nbHeuresParJour**.

Ce tableau est trié par ordre alphabétique sur le nom de l'ouvrier.

Remarque : un ouvrier peut travailler chez plusieurs clients.

1. On vous demande d'écrire le module qui affiche :
 - pour chaque ouvrier :
 - son nom,
 - la liste des clients chez qui il a travaillé (nom + total des heures prestées),
 - son nombre total d'heures prestées,
 - le nom de l'ouvrier qui a presté le plus grand nombre d'heures.
2. On vous demande d'écrire le module qui affiche pour chaque client :
 - son nom,
 - la liste des ouvriers qui y ont travaillé.

Partie 3

Objectif : comprendre et utiliser des algorithmes de tri

Exercice 8

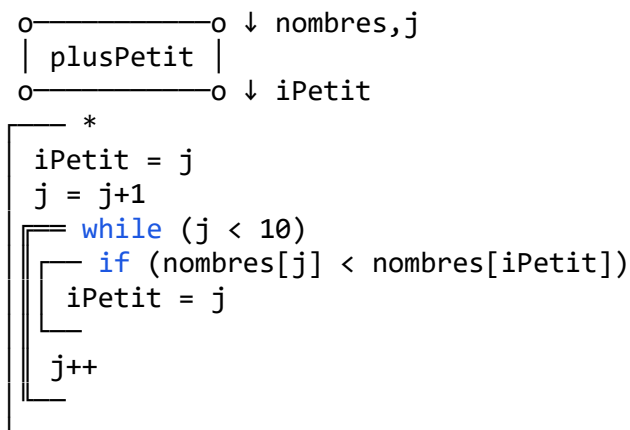
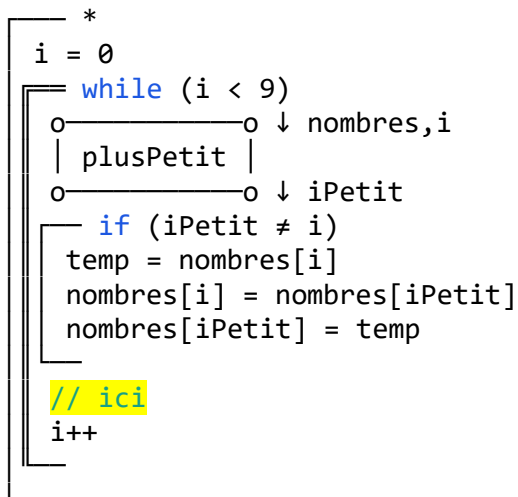
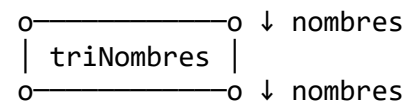
Soit le tableau nombres de 10 cellules initialisé comme suit :

13	22	57	41	3	16	80	100	94	70
----	----	----	----	---	----	----	-----	----	----

On applique à ce tableau l'algorithme de tri ci-dessous.

1. Donnez une description du contenu du tableau pour les 5 premières fois où l'on passe à l'endroit marqué **//ici**.

2. Précisez les valeurs affichées si l'on ajoute l'instruction `sortir nombres[i]` à l'endroit marqué **//ici**.



Exercice 9

On applique l'algorithme de tri à bulles ci-dessous sur le même tableau qu'à la question précédente.

1. Donnez une description du contenu du tableau pour les 5 premières fois où l'on passe à l'endroit marqué **//ici**.
2. Combien de fois le programme va-t-il passer par l'endroit marqué **//ici** ?

o ————— o ↓ nombres
| tabTriéParTriBulles |
o ————— o ↓ nombres

```
*
doitPermuter = true
while (doitPermuter)
    i = 0
    doitPermuter = false
    while (i < 9)
        if (nombres[i] > nombres[i+1])
            temp = nombres[i]
            nombres[i] = nombres[i+1]
            nombres[i+1] = temp
            doitPermuter = true
        i++
    // ici
```