

M1103 : Structures de Données & Algorithmes fondamentaux Feuille TP n° 2

Algorithmes classiques sur des tableaux - tris

OBJECTIFS PEDAGOGIQUES :

- 1.- Coder les algorithmes sous forme modulaire.
 - 2.- S'exercer à l'écriture progressive de programmes.
 - 3.- Se familiariser avec le principe de fabrication d'une bibliothèque de sous-programmes spécialisée. Ici, la bibliothèque a pour nom `bibliothequeTableaux`.
-

DOCUMENTS A VOTRE DISPOSITION POUR REALISER CE TP :

- `tp2.pdf` : la présente feuille de tp n°2 contenant le sujet du TP.
- **ressourcesTP2.zip** : une archive composée de :
 - un fichier de spécification (.h) et de définition (.cpp) d'une bibliothèque **bibliothequeTableaux** dont le but est de proposer (aux programmeurs) un certain nombre d'opérations sur des tableaux contenant des éléments *entiers ou bien des éléments complexes*. Ils seront à compléter.
 - un fichier **definitionsTypes.h** contenant les définitions des types complexes gérés par les tableaux de la bibliothèque
 - le code source du programme principal testant le tri de la Bulle. Il vous servira de modèle pour les autres programmes de test.
 - un modèle de feuille de tests (`feuilleTests_tp2.xlsx`), format Open Office, pour réaliser les tests fonctionnels des sous-programmes à coder dans cette feuille de TP.

A CODER OBLIGATOIREMENT

- **Code 1** : Le tri Bulle d'un tableau d'entiers
- **Code 2** : Le tri Bulle d'un tableau d'éléments de type `UnePersonne`, selon le nom, selon la ville
- **Code 3** : Le tri par Sélection de Places d'un tableau d'entiers, vu en cours.

Vous disposez de 1 séance de TP pour réaliser ce travail. Si vous avez terminé avant la fin du temps imparti, nous ne pouvons que vous recommander de faire aussi le Tri par Insertion.

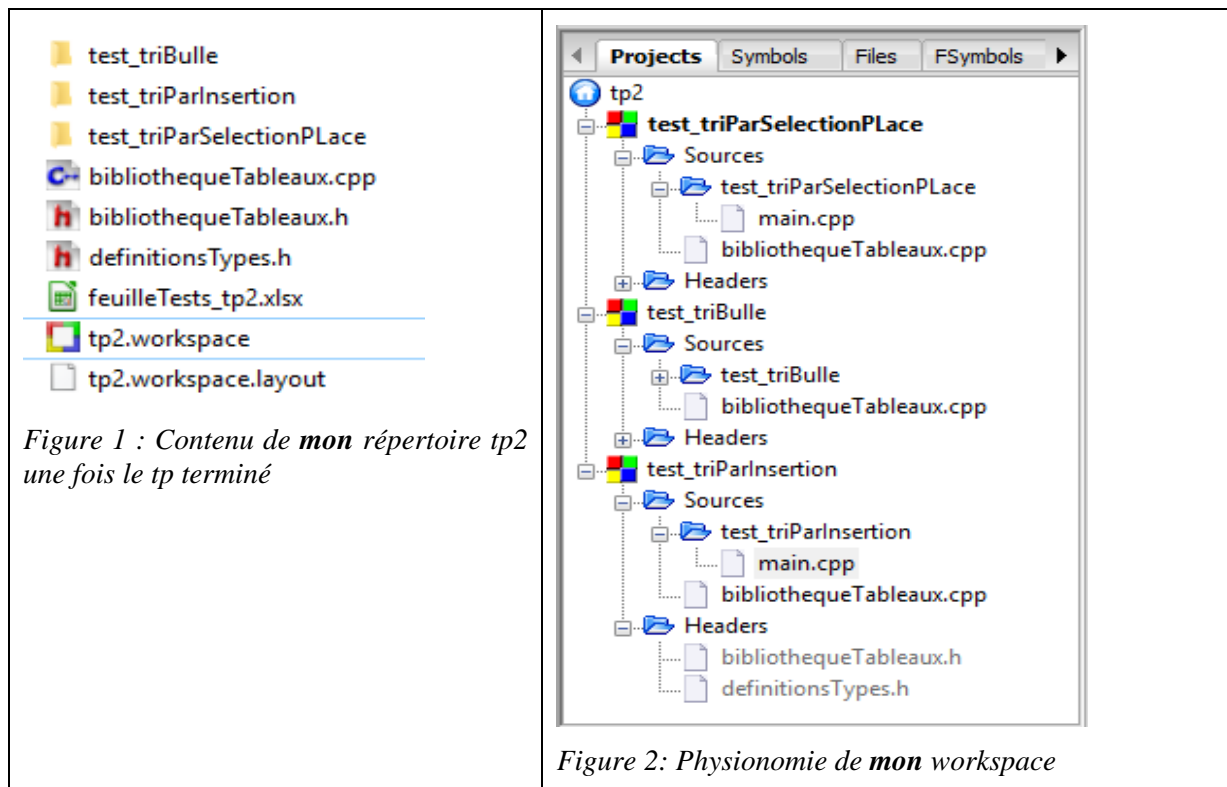
DIRECTIVES GENERALES

1. Avant de continuer, lire le contenu de la feuille de TP pour prendre connaissance du travail à faire.

DIRECTIVES PARTICULIERES POUR CETTE SEANCE DE TP

2. Dans votre répertoire consacré aux TPs du module M1103, créer un répertoire TP2. Il contiendra tous les exercices qui vous sont demandés dans cette feuille de TP.
3. Dans le répertoire `tp2` :
 - Vous devrez créer un projet par programme à écrire (mais attendez la suite du TP pour le faire)
 - Vous regrouperez ensuite tous les projets dans un même workspace nommé `tp2` (mais attendez la suite du TP pour le faire)
4. Dézipper l'archive `ressourcesTP2.zip` fournie et placer tous les fichiers .h et .cpp de l'archive à la **racine du répertoire TP2**.

À titre d'illustration, voici ce que contient mon répertoire `tp2` et mon Workspace une fois le TP **terminé** :



Code 1.- Tri Bulle sur un tableau d'entiers

5. Dans le répertoire TP2, créer un projet **test_triBulle**
6. Ajouter au projet les fichiers **bibliothequeTableaux.h**, **bibliothequeTableaux.cpp** et **definitionsTypes.h**
7. Remplacer la totalité du code du fichier **main.cpp** de votre projet par celui du fichier **main.cpp** qui vous a été fourni.
8. Vous pouvez supprimer le fichier **main.cpp** situé à la racine de votre répertoire TP2, ou encore le garder comme modèle pour les autres programmes de test du TP. Vous le supprimerez alors une fois tous les programmes écrits.
9. Renommer le workspace (tp2) et enregistrer le nouveau nom.
10. Compiler pour vérifier que les liens entre les fichiers sont corrects avant le démarrage du codage.
11. Analyser les fichiers qui vous ont été fournis :
 - Dans quel fichier est défini le type **UnePersonne** ?
 - Dans le fichier **bibliothequeTableaux.h**, comment se fait-il qu'il y ait 2 sous-programmes **afficherTableau** ou **echanger** ? Font-ils la même chose ? Quel est le nom de cette propriété ?
 - Expliquer la structure en 3 zones du fichier **bibliothequeTableaux.cpp**
12. Exécuter le programme de test (**main.cpp**). Commentez les résultats affichés à l'écran par rapport au travail à faire pour écrire le **Code 1**.
13. Quel fichier et quel sous-programme devez-vous compléter pour faire le travail demandé ?
14. Compléter le code du sous-programme identifié.
15. Préparer la feuille de tests avec des jeux d'essais pertinents, c'est-à-dire couvrant tous les cas de figures qui pourront tester la robustesse de votre algorithme/code.
16. Reporter les résultats d'exécution de votre programme sur la feuille de tests. Corriger si nécessaire et procéder à un nouveau test jusqu'à ce que tous les tests soient ok.

Code 2.- Deux tris Bulle pour un tableau d'éléments complexes (selon le nom, selon la ville)

17. Toujours dans le programme de test, dé-commenter l'instruction suivante :

```
/* Question 17
   afficherTableau(monAgenda, TAILLE);
*/
```

Compiler et exécuter à nouveau le programme de test. Commenter les résultats affichés.

18. Tri Bulle croissant selon le nom de la personne : dans quel fichier et quelle(s) numéros de lignes devez-vous écrire le code du sous-programme demandé ? Comment modifier le programme de test (**main.cpp**) pour qu'il teste votre sous-programme ?
19. Coder le sous-programme de tri Bulle selon le nom. Préparer la feuille de tests. Reporter les résultats de vos tests sur la feuille de tests.
20. Tri Bulle croissant selon la ville où habite la personne : dans quel fichier et quelle(s) numéros de lignes devez-vous écrire le code du sous-programme demandé ? Comment modifier le programme de test (**main.cpp**) pour qu'il teste votre sous-programme ?
21. Coder le sous-programme de tri Bulle selon la ville. Préparer la feuille de tests. Reporter les résultats de vos tests sur la feuille de tests.
22. Analyser les résultats produits par l'application d'un tri Bulle (sur le nom) suivi du tri Bulle (sur la ville) au tableau **monAgenda**. Que constatez-vous ? Que peut-on en conclure sur la stabilité du tri Bulle ?

Code 3.- Le tri par Sélection de Places d'un tableau d'entiers, vu en cours

23. Dans le répertoire TP2, créer un projet **test_triParSelectionPlace**
24. Ajouter au projet les fichiers **bibliothequeTableaux.h**, **bibliothequeTableaux.cpp** et **definitionsTypes.h**
25. Inspirez-vous du fichier **main.cpp** qui vous a été fourni pour produire rapidement le contenu du fichier **main.cpp** de ce projet.
26. Compléter les fichiers **bibliothequeTableaux.h** et fichiers **bibliothequeTableaux.cpp** avec le code des sous-programmes composant le **triParSelectionDePlace** vu en cours.
27. Préparer la feuille de tests. Reporter les résultats de vos tests sur la feuille de tests.
-
28. Si vous vous voulez encore vous exercer, vous pouvez faire appliquer ce tri au tableau **monAgenda** afin de découvrir si le tri par Sélection de place est stable ou pas. Vous pouvez également coder le tri par Insertion. Vous pouvez aussi, pour chacun des tris, compter puis afficher le nombre d'accès au tableau.

RAPPEL DES PRINCIPALES BONNES PRATIQUES VUES JUSQU'À MAINTENANT

- Écriture progressive du code en validant chaque étape par une opération de compilation ;
- Respect des règles de nommage des variables et des constantes ;
- Chaque variable déclarée est accompagnée d'un commentaire indiquant son rôle. Ce commentaire est écrit au moment où on déclare la variable et non à la fin une fois que le programme est terminé . . .
- Chaque variable est définie par le type qui la représente au mieux : unsigned short int pour un pourcentage plutôt qu'un simple int par exemple ;
- Le code doit toujours être indenté ;
- Les structures de contrôles sont toujours écrites en deux étapes : écriture du squelette de la structure (soit manuellement, si possible en privilégiant les abréviations de Code::Blocks) puis remplissage de la structure ;
- Les paramètres des sous-programmes que vous écrivez devront répondre aux bonnes pratiques vues en cours et résumées dans les documents du module M1102 disponibles sur le webcampus.

Pensez à intégrer chacune de ces bonnes pratiques dès que vous codez et surtout, sollicitez votre enseignant pour qu'il vous donne un avis sur les codes que vous avez terminés.

Ne codez jamais un exercice sans avoir, au préalable, réalisé l'algorithme correspondant.

Même en TP, si vous devez écrire un code dont vous n'avez pas l'algorithme, lâchez le clavier, prenez une feuille et un crayon et concevez votre solution sur papier puis faites la valider par votre enseignant.

Vous demandez à vos enseignants ce que vous ne comprenez pas.

Aucune aide ne vous sera fournie en TP si vous n'êtes pas en mesure de montrer l'algorithme sur lequel vous appuyez pour élaborer votre code.