

## Feuille de TD-TP n°7 de Programmation 2 (Tableaux de d'adresses, recherches et tris)

### Exercice1 (TD). Tableaux d'adresses d'opéras : recherches et tris

On définit les trois structures ci-dessous.

```
1 struct date {
2     unsigned jour;
3     unsigned mois;
4     unsigned annee;
5 };
6
7 struct individu {
8     char * nom; /* nom de l'individu */
9     char * prenom; /* prenom de l'individu */
10    date * naissance; /* date de naissance de l'individu */
11 };
12
13 struct opera {
14     char * titre; /* titre de l'opera */
15     date * date_creation; /* annee de la creation (i.e. de la 1re representation) */
16     char * ville_creation; /* ville de la creation (i.e. de la 1re representation) */
17     individu * compositeur; /* nom du compositeur */
18 };
```

Dans la suite de cet exercice, on appellera « tableau d'opéras » un tableau dont les clés sont des adresses d'opéras.

Vous supposerez écrite une fonction `void afficher_opera(const struct opera * op)` qui affiche les informations importantes sur l'opéra pointé par `op` (compositeur, titre, date et ville de création).

a) Écrivez la définition d'une fonction `age_revolu` qui reçoit en arguments un individu `ind` et une date `d` et renvoie le nombre d'années révolues de `ind` à la date `d`. Par exemple, un individu né le 7 avril 1950 a 49 années révolues le 7 avril 2000 et en a 50 le 8 avril 2000.

b) Écrivez la définition d'une fonction `recherche_tab_opera_age` qui reçoit en arguments un tableau d'opéras `tab`, sa taille `n` et un entier positif `a` et affiche tous les opéras du tableau `tab` dont la première représentation a eu lieu avant que leur compositeur ait `a` années révolues; cette fonction ne renvoie pas de valeur. Combien d'opérations arithmétiques, de comparaisons et d'affectations sont-elles exécutées quand on appelle la fonction telle que vous l'avez écrite? Évaluez cette grandeur en fonction (des valeurs) des arguments.

c) Écrivez la définition d'une fonction `trier_tab_opera_creation` qui reçoit en arguments un tableau d'opéras `tab` et sa taille `n` et trie le tableau selon **l'ordre chronologique de première représentation des opéras**. La fonction ne renvoie pas de valeur; elle implémente un **tri par sélection**.

d) Écrivez la définition d'une fonction `reorganiser_tab_opera_ville` qui reçoit en arguments un tableau d'opéras `tab` et sa taille `n`, et une chaîne de caractères `v`, et réorganise le tableau en plaçant en tête du tableau les opéras dont la première représentation a eu lieu dans la ville `v`. La fonction renvoie le nombre d'opéras de `tab` créés dans la ville `v`. Combien d'opérations arithmétiques, de comparaisons et d'affectations sont-elles exécutées quand on appelle la fonction telle que vous l'avez écrite? Évaluez cette grandeur en fonction (des valeurs) des arguments.

e) Écrivez la définition d'une fonction **récursive** `recherche_dicho_tab_opera_date` qui reçoit en arguments un tableau d'opéras `tab` (**supposé trié dans l'ordre chronologique de première représentation des opéras**)

et sa taille  $n$ , et un entier positif  $a$ , et affiche un opéra dont la première représentation a eu lieu pendant l'année  $a$  s'il s'en trouve un; la fonction affiche un message d'absence si aucun opéra de *tab* n'a été donné pour la première fois pendant l'année  $a$ . La fonction ne renvoie pas de valeur; elle implémente une **recherche dichotomique**.

f) Écrivez la définition d'une fonction `trier_tab_opera_compositeur` qui reçoit en arguments un tableau d'opéras *tab* et sa taille  $n$  et trie le tableau selon l'**ordre lexicographique des noms de compositeur**. La fonction ne renvoie pas de valeur; elle implémente un **tri par insertion**.

## Exercice 2 (TP). Manipulation de tableaux d'adresses d'opéras

Créez un répertoire TD-TP7 et téléchargez-y les deux fichiers du dossier « Fichiers pour le TP n°7 » déposé sur le site Moodle du cours :

- un fichier `opera.c` contenant des définitions de fonctions utiles pour le TP;
- un fichier `texte Operas_entree.txt`.

**Le fichier `Operas_entree.txt` ne doit pas être modifié**; dans un terminal, depuis le répertoire TD-TP7 dans lequel vous l'avez téléchargé, lancez la commande

```
chmod ugo-w Operas_entree.txt
```

pour empêcher toute modification du fichier.

Écrivez un programme qui

1. lit le nom de fichier qui lui est passé comme argument depuis la ligne de commande et utilise le fichier ainsi nommé pour initialiser un tableau d'opéras `tab_operas`;
2. affiche tous les opéras de `tab_operas`;
3. affiche les opéras de `tab_operas` créés avant que leur compositeur ait quarante ans révolus;
4. affiche tous les opéras de `tab_operas` après avoir réordonné le tableau selon l'ordre chronologique des créations;
5. recherche un opéra créé en 1831 puis un opéra créé en 1900;
6. affiche tous les opéras de `tab_operas` après avoir réordonné le tableau selon l'ordre lexicographique (c'est-à-dire l'ordre des mots dans le dictionnaire) des noms de compositeur;
7. affiche tous les opéras de `tab_operas` après avoir placé tous les opéras créés à Paris au début du tableau;
8. libère avant de terminer son exécution tout l'espace alloué sur la tas.

### Consignes et conseils pour l'écriture du programme.

- Pour l'étape 1, vous utiliserez la fonction `initialiser_tab_opera_fichier` définie dans le fichier `opera.c`; lisez attentivement le commentaire qui figure avant sa définition; ne perdez pas de temps à essayer de comprendre la définition, ce sera l'objet d'un cours futur !
- Pour certaines des étapes 2 à 8, vous pourrez utiliser les fonctions définies dans `opera.c`. Lisez les commentaires qui précèdent leur définition pour savoir comment les utiliser.
- N'oubliez pas d'inclure la bibliothèque d'entêtes `<string.h>` au début de votre programme.
- Après l'avoir compilé sans erreur, testez votre programme en lui passant comme argument le nom du fichier `Operas_entree.txt`.