ECOLE CENTRALE DE NANTES

SMP - TP4

1 INTRODUCTION

L'objectif de ce TP est de créer et de manipuler concrètement ces structures de données et de comparer les temps d'exécution des opérations de base, à savoir l'ajout, la recherche, la suppression ou l'affichage d'éléments.
Le thème du TP est l'organisation d'informations relatives à des personnes à travers la gestion d'un répertoire contenant des éléments de type `personne`, lui même contenant les informations relatives à une personne. Ce répertoire est représenté avec **deux structures de données différentes** : tableau trié et liste triée. Ces structures de données sont ordonnées d'abord selon le nom de la personne, puis son prénom et enfin son numéro de téléphone.

2 STRUCTURES DE DONNÉES

Définissez les structures suivantes et implémentez-les dans un fichier `type_def.h` :

QUESTION 1. `personne` contenant le nom (chaine de caractère), le prénom (chaine de caractère), un numéro de téléphone (10 chiffres dans une chaine de caractère sans espaces).

QUESTION 2. `elementListe` un enregistrement contenant une `personne` et les pointeurs permettant de créer une liste chainée (suivant) (bonus : doublement chainée (suivant, précédent)).

3 UTILITAIRES

Définissez les fonctions suivantes et implémentez-les dans le fichier `utilitaires.cpp`. Vous testerez vos fonctions avec un programme principal (`main`) dans un fichier `TP4.cpp`.

QUESTION 3. `genererPersonne` : Génération aléatoire d'une personne à partir des fonctions `genererNomPrenom` et `genererTel` fournies dans `utilitaire_generation.cpp`. Un exemple d'utilisation de ces fonctions est donné dans le fichier `main.cpp` fourni. Cette fonction renvoie une `personne`.

QUESTION 4. `creerElementListe` : Création d'un `elementListe` dont la valeur est de type `personne`

QUESTION 5. `affichagePersonne` : affichage à l'écran du nom, prénom et numéro de téléphone d'une personne passée en paramètre.

QUESTION 6. `egalitePersonne` : Détermine si deux personnes ont les mêmes informations, nom, prénom et numéro de téléphone. Cette fonction retourne un booléen : vrai en cas d'égalité, faux dans l'autre cas.

QUESTION 7. `comparerPersonne` : Définition d'un ordre sur les personnes (qui sera utilisé pour ordonner les structures de données). Cette comparaison se fait d'abord selon le nom de la personne, puis son prénom et enfin son numéro de téléphone. On considère que les deux personnes ne sont pas identiques en entrée de la fonction. Cette fonction retourne un booléen : vrai si la première personne est avant l'autre, faux dans l'autre cas.

Indice pour les Q6 et Q7 : Les `string` peuvent se comparer avec les opérateurs de comparaison standard (==, !=, <, <=, >, >=)

4 CRÉATION ET UTILISATION D'UN RÉPERTOIRE

Définissez les fonctions suivantes pour les listes et les tableaux et implémentez-les dans un fichier `repertoire.cpp`.

QUESTION 7. `ajouter` : Ajout d'une personne au bon endroit dans la structure de données (liste ou tableau) en utilisant la fonction de comparaison. Si on trouve une personne identique à une autre déjà présente dans la liste, la personne n'est pas ajoutée. Pour une liste, cette fonction renvoie un pointeur vers la tête de la nouvelle liste. Cette fonction va vous permettre de créer des répertoires de personnes triées.

QUESTION 8. `afficher` : Affichage de la structure de données (liste ou tableau) dans l'ordre.

QUESTION 9. `rechercher` : Recherche d'une personne dans la structure de données (liste ou tableau) à partir du nom, prénom et numéro de téléphone. Cette fonction renvoie l'indice de la personne trouvée, sinon -1.

QUESTION 10. `supprimer` : Suppression d'une personne dans la structure de donnée (liste ou tableau) à partir du nom, prénom et numéro de téléphone. Pour les listes cette fonction renvoie un pointeur vers la tête de la nouvelle liste.

5 TESTS ET RAPPORT

Vous créez des répertoires de 1000 personnes et vous comparerez les temps d'exécution entre les structures de données liste et tableau pour

- la création des répertoires
- l'affichage des répertoires
- la recherche de 100 personnes
- la suppression de 100 personnes

Les temps d'exécution peuvent être comparé avec la librairie `ctime`, en particulier avec la fonction `clock` : `http://www.cplusplus.com/reference/ctime/clock/`

Vous mettrez dans le rapport toutes les structures de données, les spécifications des fonctions utilitaires, ainsi que les algorithmes des fonctions d'ajout, affichage, recherche et suppression.
Vous indiquerez les résultats des tests et commenterez le comportement de vos fonctions dans les cas normaux et les cas limites.
Vous déposerez vos codes et comptes rendus sur hippocampus.