

Devoir 1
Structure de données (8INF259)

**Ce devoir doit être remis au plus tard
le mercredi 8 octobre avant minuit**

Instru Instructions

- * Pour faciliter la correction de vos programmes, il est recommandé de bien commenter vos programmes.
- * Travaillez en équipe **au plus de deux étudiant(e)s**, remettez une seule copie par équipe.
- * Il est interdit pour une équipe de copier le travail d'une autre équipe.
- * Votre chargé de TD est Yao Jean-Eudes Adjahoun. Son adresse courriel est :
yajadjahoun@gmail.com

=====

Objectifs du travail demandé

Le principal but de ce travail est de se familiariser avec les pointeurs et les classes dans le langage C++. Il est donc demandé dans ce TP de concevoir un programme, écrit en C++, créant une liste chaînée en mémoire à partir de données contenues dans un fichier texte **PF**. Ce dernier contient de l'information concernant des dossiers de professeurs. Chaque dossier de professeur contient son nom, ancienneté, une liste de cours qu'il souhaite enseigner et une liste de noms d'étudiants qu'il souhaite diriger. Pour ce faire, il est demandé de programmer une classe **DossierProfesseur** qui crée, à partir des données contenues dans le fichier texte **PF**, une liste chaînée (dont la structure est décrite plus loin). Cette classe permettra d'effectuer les opérations suivantes sur cette liste :

1. Supprimer un dossier professeur donné.
2. Afficher le nom du professeur ayant le plus d'étudiants. En cas de réponses multiples, choisir le premier professeur le moins ancien.
3. Afficher le cours le plus demandé. En cas de solutions multiples, choisir celui donné par le (premier) professeur le moins ancien apparaissant dans la liste des professeurs.
4. Afficher le nombre de professeurs souhaitant enseigner un cours donné.
5. Recopier la liste chaînée mise à jour dans autre fichier Filename.

Les informations contenues dans le fichier texte FP ont le format suivant :

```
nom
ancienneté
cours 1
cours 2
.
.
cours n
& // ici & sert à séparer la liste des cours de celle des étudiants
Etudiant 1
Etudiant 2
.
.
& //cette fois-ci & sert à passer au professeur suivant
```

Les noms sont des chaînes de caractères identifiant un professeur ou un étudiant. Les cours 1, cours 2, ... etc sont des chaînes de caractères identifiant des sigles de cours. Utiliser les sigles suivants: INF100, INF101, INF102, INF103, ..., etc. Les valeurs de Etudiant 1, Etudiant 2, ..., etc sont des chaînes de caractères identifiant le nom des étudiants que souhaite diriger le professeur en question. Le caractère & sert la première fois à séparer les sigles de cours des noms des étudiants et la deuxième fois à séparer les dossiers de professeurs.

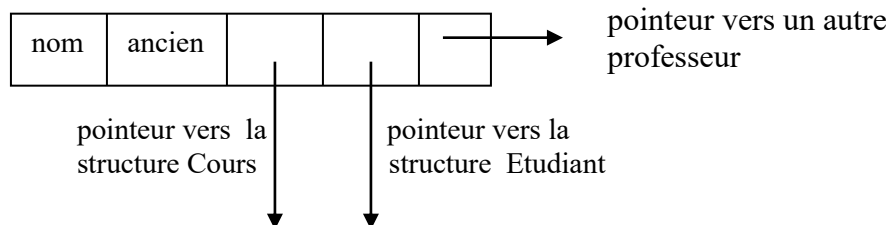
Votre programme lira à partir d'un autre fichier texte FT, les transactions à effectuer par votre programme dont le format est: op [param1]. Les opérations possibles sont comme ci-dessous.

- name : Supprimer tous les professeurs dont le nom est name de la liste chaînée.
- # : Afficher le nom du professeur ayant le moins d'étudiants dans la liste chaînée.
- * : Afficher le cours le moins demandé dans la liste chaînée.
- % coursename : Afficher le nombre de professeurs souhaitant enseigner le cours coursename.
- \$: Recopie la liste chaînée mise à jour dans le fichier FP.

Les structures de données à utiliser sont les suivantes :

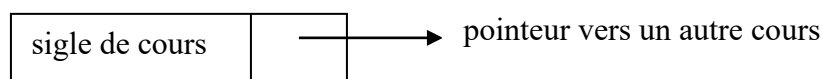
```
struct Professeur{
    string nom;
    int ancien;
    Cours* listecours;
    Etudiants* listetudiants;
    Professeur* suivant;
};
```

Graphiquement, cette structure est comme suit :



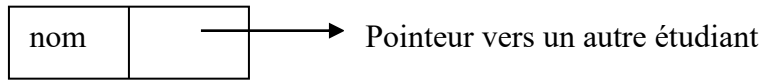
```
struct Cours{
    string sigle;
    Cours * suivant;
};
```

Graphiquement, cette structure est comme suit :



```
struct Etudiant {
    string nom;
    Etudiant* apres;
};
```

Graphiquement, la structure Etudiant est comme suit :



```
classe DossierProfesseur {
```

```
private:
```

```
    Professeur *tete; // début de la liste chaînée
```

```
public:
```

```
Dossierprofesseur (char* PF); // Construit de la structure de la liste chaînée en mémoire à partir
                                // du fichier PF
```

```
~Dossierprofesseur ( ); // Détruit de la liste chaînée existant en mémoire.
```

```
void supprimer (char *name); // Supprime de la liste chaînée tous les professeurs dont le nom est name
```

```
char* afficher leprofplusetudiant ( ) const; // Affiche le premier professeur ayant le plus d'étudiants
```

```
char* affichercoursplusdemande ( ) const; // Affiche le cours le plus demandé de la liste. En cas de
                                           // de litige, choisir celui correspondant au (premier) professeur
                                           // le moins ancien apparaissant dans la liste.
```

```
int affichernbreprofpouruncours (string * coursdonne) const; // affiche le nombre de professeurs pour le
                                                             // cours coursdonne.
```

```
void recopier (char *PF); //recopie la liste chaînée dans le fichier PF.
```

```
};
```