

Devoir 1
Structure de données et algorithmes
(8SIF109)

**Ce devoir doit être remis au plus tard
le vendredi 13 février 2015 avant 15h45.**

Instructions

- * Pour faciliter la correction de vos programmes, il est recommandé de bien commenter vos programmes.
- * Dans votre compte sur le sunens, créez un répertoire dont le nom est Devoir1-8SIF109.
- * 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.
- * Vous devriez remettre une feuille où les informations suivantes sont indiquées:
 - 1) Le nom et prénoms des co-équipiers
 - 2) Le nom d'utilisateur du compte sur le sunens où se trouve votre code source
 - 3) Le mot de passe pour accéder à ce compte.
- * Les rapports sont à déposer dans la boîte aux lettres du cours (**8SIF109: programmation, algorithmes et structures de données**) située juste à l'extérieur de la direction du DIM au 4^{ème} étage.
- * Important : ne modifiez plus vos programmes après la date limite puisque cela changera la date de la dernière modification de vos fichiers. Par ailleurs, n'oubliez pas d'interdire l'accès à vos fichiers. Vous êtes seul(e) responsable de votre compte.

=====

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 FP. Ce dernier contient de l'information concernant des dossiers de professeurs. Chaque dossier de professeur contient son nom, son prénom, 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 FP, 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.
2. Afficher le nom de l'enseignant ayant le plus d'étudiants.
3. Afficher le cours le plus demandé.
4. Afficher le nombre des 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
prenom
cours 1
cours 2
.
.
cours n
& // ici & sert à séparer les cours des étudiants
Etudiant 1
Etudiant 2
.
.
& //cette fois-ci & sert à passer au professeur suivant

```

Le nom et prenom sont des chaînes de caractères identifiant un professeur donné. Les de cours 1, cours 2, ... etc sont des chaînes de caractères identifiant des sigles de cours. Utiliser les sigles suivants: SIF100, SIF101, SIF102, SIF103, ..., 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.

- nom, prenom : supprimer un professeur de la liste chaînée.
- # : afficher le nom du professeur ayant le plus d'étudiants dans la liste chaînée.
- * : afficher le cours le plus 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.

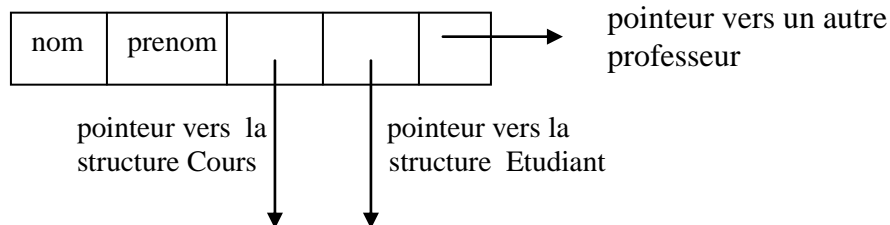
Les structures de données à utiliser sont les suivantes :

```

struct Professeur{
    char* nom;
    char* prenom;
    Cours* listecours;
    Etudiants* listetudiants;
    Professeur* suivant;
};

```

Graphiquement, cette structure est comme suit :

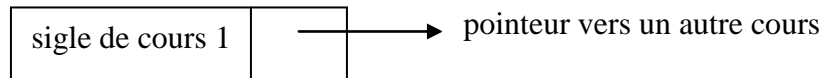


```

struct Cours{
    char    * sigle;
    Cours  * suivant;
};

```

Graphiquement, cette structure est comme suit :

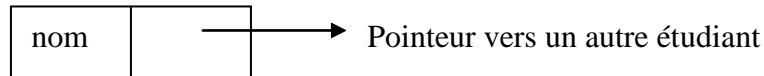


```

struct Etudiant {
    char*    nom;
    Etudiant* apres;
};

```

Graphiquement, la structure Etudiant est comme suit :



```

classe DossierProfesseur {

```

private:

```

    Professeur * tete;    // début de la liste chaînée

```

public:

```

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

```

```

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

```

```

    void supprimer (char * nom, char * prenom ); // supprime de la liste chaînée un professeur donné

```

```

    char* afficher leprofplusetudiant ( ) const;    // affiche le nom du professeur ayant le plus d'étudiants

```

```

    char* affichercoursplusdemande ( ) const;    // affiche le cours le plus demandé de la liste

```

```

    int affichernbreprofpouruncours (char* coursdonne) const; // affiche le nombre de professeurs pour le
                                                                // cours coursdonne recopie la liste chaînée

```

```

    void recopier (char *FP);          // dans le fichier FP

```

```

};

```