

TP8 (2 séances)

Objectif : programmer une classe « tabMot » permettant d'analyse statistiquement un texte.

Détails des fonctionnalités :

Lire le texte à partir d'un fichier.

Compter le nombre de mots distincts d'un texte ; et pour chaque mot, précisez son nombre d'occurrences.

Afficher la liste des mots par ordre alphabétique, avec leur nombre d'occurrences.

Afficher la liste des mots dans l'ordre de leur importance dans le texte. C'est à dire en premier les mots qui sont le plus présents dans le texte.

Compter le nombre d'occurrences de chaque lettre dans le texte.

Afficher la liste de lettres par ordre alphabétique avec leur nombre d'occurrences dans le texte.

Afficher la liste des lettres dans l'ordre de leur présence dans le texte.

Stocker le résultat de l'analyse dans un fichier.

Première étape : Lire le texte stocké dans un fichier. Le nom du fichier est donné sur la ligne de commande (paramètre de l'exécutable).

Indice codez, compilez et testez le code suivant de main.cc

```
#include <iostream>
#include<cstdlib>
using namespace std;
int main(int argc, char *argv[])
{
    cout << "=====" << endl;
    cout <<"liste des " << argc << " arguments" << endl;
    cout << "=====" << endl;
    for(int i=0; i<argc; i++)
        cout << "argv[" << i << "]" :< " << argv[i]<< endl;
    exit(0);
}
```

Testez le programme main.cc en exécutant les lignes de commande suivantes :

- main
- main toto
- main blabla bla blabla

Deuxième étape : manipuler des chaînes de caractères (objets de la classe string).

Indice : étudiez, compilez et testez le programme chaine.cc de

/net/exemples/AS_2009_2010/TP_AS_D_PROG

Pour plus d'information voir le chapitre 12 (classe string) du livre en ligne [Programmation C++ \(débutant\)](#) sur le site <http://fr.wikibooks.org/wiki/Accueil>.

Algorithmique

1. Complétez la structure statique donnée ci-dessous permettant de manipuler un texte ayant au plus NB_MAX_MOTS mots.
2. Ecrivez un algorithme affichant sur la sortie standard la liste des mots par ordre alphabétique, avec leur nombre d'occurrences.
3. Ecrivez un algorithme affichant sur la sortie standard la liste des mots par dans l'ordre de leur importance dans le texte.
4. Proposez un format de stockage des résultats dans un fichier.

Programmation

1. Complétez les fichiers d'entête « Mot.h » et « TabMot.h » contenant respectivement les déclarations des constructeurs, des destructeurs, des méthodes de la classe Mot et de la classe TabMot, ainsi que l'ensemble de leurs membres.
2. Ecrivez le code de la classe Mot et de la classe TabMot.

Dossier à rendre début janvier 2010 :

1. Présentation de votre structure de données, permettant d'analyser un texte
2. L'algorithme affichant sur la sortie standard la liste des mots dans l'ordre alphabétique, avec leur nombre d'occurrences.
3. L'algorithme affichant sur la sortie standard la liste des mots dans l'ordre d'importance, avec leur nombre d'occurrences
4. Le code facilement lisible (attention aux noms des méthodes, aux noms des variables, aux commentaires sur les méthodes, ...).
5. Jeu de tests que vous avez utilisé pour tester et valider votre code.
6. Un état honnête d'avancement du projet (précisant les fonctionnalités qui n'ont pas été implémentées, si vous avez détecté des bugs, ...).

Optionnel :

1. **Structure dynamique-** Concevez et programmez la classe « DynamiqueTabMots » permettant d'analyser des textes de taille quelconque. Vous pouvez, si vous le souhaitez, utiliser la STL (standard library of C++), mais cela n'est pas obligatoire.

```
=====
Mot.h
=====
class Mot {
private:
    string _mot;
    int _nb;
public :
    Mot() ;
    Mot(const string &str) ;
    void incrNbOccurrence() ;
    const string &getMot() ;
    int getNbOccurrence() ;
};
```

```
=====
TabMot.h
=====
class TabMots {
private :
    int _taille;
    Mot _tabMot[NB_MAX_MOTS];
public :
    TabMots() ;
    TabMot(string texte[] , int taille) ;
    TabMot(ifstream &ficEntree) ;
    void afficherAlphabetique () ;
    void afficherImportance () ;
    int getTaille() ;
    const string &getMot(int position) ;
    int getNbOccurrence(int position) ;
};
```