Rappel sur les bases

- Les variables, les opérateurs et les opérations
- Structures de contrôles
- Tableaux et pointeurs
- Fonctions et récursivité
- Quelques algorithmes de tri et de recherche
- **Fichiers**
- Structures
- Listes chainées, Piles, Files

Rappel

Fichiers

Jusque-là, nous ne savons que lire et écrire sur une console. Dans cette section, nous allons apprendre à interagir avec les fichiers.

- Par définition, un fichier est une suite d'informations stocker sur un périphérique (disque dur, clé USB, CDROM, etc. …).
- On peut accéder à un fichier soit en lecture seule, soit en écriture seule ou soit enfin en lecture/écriture.
- Pour pouvoir manipuler les fichiers en C++, il va falloir inclure la bibliothèque fstream (#include <fstream>) qui signifie "file stream" ou "flux vers les fichiers" en français.
- **Il existe 2 types de fichiers**:
  - les fichiers textes qui contiennent des informations sous la forme de caractères. Ils sont lisibles par un simple éditeur de texte.
  - les fichiers binaires dont les données correspondent en général à une copie bit à bit du contenu de la RAM. Ils ne sont pas lisibles avec un éditeur de texte.

Rappel

Fichiers: Manipulation des fichiers textes

Lorsqu'on inclut fstream, il ne faut pas inclure iostream car ce fichier est déjà inclut dans fstream.

- Lecture d'un fichier texte:
  - Pour ouvrir un fichier en lecture, la syntaxe est la suivante:

    ifstream nom_fichier ("chemin_vers_le_fichier");
  - Pour savoir si le fichier a bien été ouvert en lecture, la méthode is_open() est utilisée. Elle renvoie true si le fichier est effectivement ouvert.

    Ex: nom_fichier.is_open();
  - Pour fermer le fichier, on fait: nom_fichier.close();
  - Pour tester si on est arrivé à la fin du fichier, on fait: nom_fichier.eof();
  - La lecture dans un fichier se fait par:

    nom_fichier >> variable1 [>> variable2>> …];
    - Ici, l'espace et le saut de ligne sont des séparateurs

Rappel

Fichiers: Manipulation des fichiers textes

Exemple:

```
# include <fstream>
# include <string>
using namespace std;
int main(void)
{
	string nom;
	string prenom;
	string tel;
	ifstream f ("data.txt"); // ouverture du fichier en lecture

	f >> nom >> prenom >> tel;
	while(!f.eof()) // tant qu'on n'est pas arrivé à la fin du fichier
	{
		cout << nom << " \t"<< prenom << " \t"<< tel << "\n"; // on affiche
		f >> nom >> prenom >> tel; // on lit les informations suivantes
	}
	f.close();

	return 0;
}
```

Rappel

Fichiers: Manipulation des fichiers textes

- Ecrire dans un fichier texte:
  - La création d'un nouveau fichier ou l'écriture dans un fichier existant se fait comme suit:
    - **ofstream** ***nom_fichier*** ("chemin_vers_le_fichier");
  - L'écriture dans un fichier se fait par:
    - **nom_fichier** <<"cheikh"<<" "<<"diop"<<" "<<"772220202"<<"\n";
      - Il va falloir écrire le séparateur soi-même.

Rappel

Fichiers: Manipulation des fichiers textes

Exemple:

```
# include <fstream>
# include <string>
using namespace std;
int main(void)
{
        string nom;
        string prenom;
        string tel;

         ofstream f ("data.txt"); // ouverture du répertoire en écriture
         for(int i=0; i<10; i++){
                cout << "\n p"<< i+1 << ":\n"
                cout << "nom:";
                cin >> nom;
                f <<nom << " ";
                cout << "\n prenom:";
                cin >> prenom;
                f <<prenom<< " ";
                cout << "\n tel:";
                cin >> tel;
                f << tel << "\n";
        }
         f.close();

        return 0;
}
```

Rappel

Fichiers

EXERCICES D'APPLICATIONS

- Application 22:

Écrire un programme qui écrit dans le fichier data.txt le texte suivant:

- Bonjour les étudiants!
- Bonjour Professeur
- Comment allez-vous?
- Nous allons bien merci et de votre coté?
- Ça va bien aussi, merci!

- Application 23:

Soit le fichier data.txt précédemment créé, écrire un programme un programme permettant de lire puis d'afficher son contenu.