

TP 06 – Manipulation de fichiers

Exercice 0 Mise en place du TP

- Créez une nouvelle solution Visual Studio comme vous en avez l'habitude.
- Récupérez sur l'ENT le fichier *VEtudiant.zip* et copiez son contenu (les fichiers *inputUTF8.csv* et *inputUTF8_v2.csv*) dans le répertoire du projet (là où se trouve le fichier *.vcxproj*)

Exercice 1 Lecture / écriture d'un fichier

Le C++ propose d'accéder aux fichiers de la même manière (ou presque) que vous accédez à la console par le biais de la bibliothèque `<fstream>`. Cette bibliothèque apporte deux nouveaux types au C++ : `std::ifstream` pour la lecture de fichiers et `std::ofstream` pour l'écriture.

On crée une variable de l'un de ces types en passant entre parenthèses le chemin d'accès au fichier. L'exemple suivant montre l'ouverture du fichier « *inputUTF8.csv* » et la vérification que l'ouverture s'est correctement déroulée.

```
std::ifstream in("inputUTF8.csv");  
if (!in.is_open())  
    std::cerr << "Problème d'ouverture du fichier \"inputUTF8.csv\".\n";
```

Code 1 : Ouverture du fichier « *inputUTF8.csv* » en lecture.

1.1 Lecture de fichiers

La lecture se fait à la manière d'une lecture depuis `cin`. Lorsque la fin du fichier ouvert par la variable `in` est atteinte, l'expression `in.eof()` renvoie `true` (EOF = End Of File).

- Dans votre programme, créez une variable permettant d'ouvrir le fichier *inputUTF8.csv* comme dans l'exemple de code 1.
- Lisez ce fichier ligne par ligne et affichez-les dans la console.

1.2 Écriture de fichiers

L'écriture dans un fichier se fait exactement comme pour `cout`.

- Ajoutez à votre programme une variable d'accès en écriture au fichier « *outputUTF8.csv* ».
- Ajoutez chaque ligne non vide du fichier *inputUTF8.csv* au fichier *outputUTF8.csv*.
- Demandez à l'utilisateur ses nom, prénom et date de naissance.
- Ajoutez une nouvelle ligne à *outputUTF8.csv* constituée de ces trois données, dans cet ordre, et séparées par des virgules.
- Ouvrez le fichier *outputUTF8.csv* dans un éditeur de texte pour vérifier.

Exercice 2 Le format de fichier CSV

Le format de fichier **CSV** (*Comma-Separated Values*) est un format simple permettant de stocker des informations en texte sous forme de tableau. Les fichiers qui vous ont été fournis sont deux exemples de ce format.

Chaque ligne d'un fichier CSV est constituée d'un certain nombre de champs séparés par des virgules. Si un champ doit contenir une virgule, il est possible de l'encadrer par des doubles guillemets.

Le fichier *inputUTF8.csv* est un exemple simple, sans usage de guillemet. Le fichier *inputUTF8_v2.csv* contient les mêmes données, mais certaines sont encadrées de guillemets doubles.

2.1 Analyse et décodage du fichier *inputUTF8.csv*

Les chaînes de caractères peuvent être elles aussi manipulées comme **cin** et **cout** ou les variables de type **ifstream** / **ofstream**. Pour cela, on crée un flux connecté à la chaîne de caractères par le biais d'une variable du type **std::stringstream** apporté par la bibliothèque **<sstream>**. L'exemple suivant montre comment interpréter une chaîne de caractères à la façon de **cin**.




```
std::string texte = "2 4 Bonjour";
std::stringstream sstr(texte);
int deux, quatre;
std::string Bonjour;
sstr >> deux >> quatre >> Bonjour;
```

Code 2 : Décodage de la chaîne de caractères *texte* à la façon d'un flux.

Enfin, on peut séparer des champs séparés par un caractère particulier en utilisant à nouveau l'instruction **std::getline(sstr, texte, ',')**; comme dans l'exemple suivant.




```
std::string texte = "Champ1,Champ2,Champ3";
std::stringstream sstr(texte);
std::string ch1, ch2, ch3;
std::getline(sstr, ch1, ',');
std::getline(sstr, ch2, ',');
std::getline(sstr, ch3, ',');
```

Code 3 : Décodage de la chaîne de caractères *texte* en récupérant trois champs séparés par une virgule grâce à **std::getline**.



-  Créez un nouveau programme qui décodera chaque ligne du fichier *inputUTF8.csv* en séparant les champs. Vous stockerez alors le fichier décodé dans un tableau à deux dimensions de chaînes de caractères.
-  Une fois le tableau constitué, vous l'afficherez en alignant correctement les données
 -  Utilisez pour cela des tabulations **'\t'** et des manipulateurs de flux comme vu dans les TP précédents.


2.2 Ajout d'un nouvel enregistrement

Comme vous l'avez sûrement remarqué, la première ligne du fichier CSV est constituée des entêtes des colonnes.


-  En utilisant **automatiquement** les noms des entêtes, demandez à l'utilisateur d'ajouter une nouvelle ligne.
-  Enregistrez les nouvelles données dans le fichier *outputUTF8.csv*.
 -  Vérifiez le fichier à l'aide d'un éditeur de texte.

2.3 Gestion des guillemets dans les champs

-  Reprenez tout l'exercice 2 en utilisant cette fois le fichier *inputUTF8_v2.csv*.
 -  Les guillemets ne font pas partie des données. Ils ne doivent pas être affichés ni stockés dans le tableau des données en mémoire.

 Ceci est un support succinct du format CSV. D'autres détails sont normalement à respecter, même s'ils sont rarement rencontrés. Le format est spécifié ici :

<https://datatracker.ietf.org/doc/html/rfc4180>

-  En cas d'insomnie, vous pouvez toujours y jeter un œil et tenter de l'implémenter.