Master Langue et Informatique

Université Paris-Sorbonne

Programmation générique et conception objet

Atelier n°2

Manipulation des fichiers et des chaînes de caractères

Les exercices qui vous sont proposés ici ont pour but d'apprendre à utiliser des fichiers binaires et textes en C++ (www.cplusplus.com/ref/iostream).

1. MON PREMIER FICHIER TEXTE EN C++

Exercice 1 : Créer le programme exécutable MonPremierFichierTexte qui écrit dans le fichier texte « Naissance.txt » votre date de naissance.

Exercice 2 : Ajouter au programme exécutable MonPremierFichierTexte la fonction *Verifier* qui vérifie si le fichier « Naissance.txt » contient bien votre date de naissance.

2. MANIPULATIONS DE FICHIERS

Le but de cette partie est de manipuler les différents types de flots de la bibliothèque de classes iostream (www.cplusplus.com/ref/iostream)

Exercice 0 : Soit un tableau **tab** de 20 entiers de type short (que vous créez manuellement), écrire un programme qui va écrire ce tableau dans un fichier binaire **tp2.bin.** Essayez d'ouvrir ce fichier avec un éditeur de texte standard.

Exercice 1 : Soit le fichier binaire « tp2.bin » contenant des entiers de type short. Ecrire un programme permettant de lire le contenu de ce fichier.

Exercice 2 : Soit le fichier binaire « tp2.bin. Écrire un programme permettant de lire le $i^{ème}$ entier. Tester votre programme avec i = 10.

Exercice 3 : Soit le fichier binaire « tp2.bin » contenant des entiers de type short. Ecrire un programme permettant de modifier le $i^{ème}$ entier. Tester votre programme avec i = 10.

Exercice 4 : Créer le programme exécutable Copie qui permet de copier le contenu d'un fichier dans un autre.

3. TOKENIZATION DE TEXTE (A DÉPOSER AVANT LE 13/03)

La *tokenization* est le processus permettant de découper une suite de caractères données en un ensemble de *tokens* (suites de caractères ayant un sens minimal pour une application donnée). Une tokenization brutale (méthode 1) consiste à utiliser les séparateurs (espace ou tabulation) pour déterminer les tokens. Une autre (méthode 2) consiste à utiliser tout ce qui n'est pas un caractère alphanumérique (ie. [^0-9A-Za-z] dans un regexp) comme séparateur.

Exercice 4 : Soit la première phrase du fichier litterature.txt

« Beaucoup d'érudits et de lettrés s'imaginent volontiers que la Belgique est une création artificielle,œuvre de l'histoire et des volontés humaines, et ne s'appuyant sur aucun fait éternel de la nature »

a) En utilisant, manuellement, les deux méthodes décrites ci-dessus, combien cette phrase comporte de tokens ?

b) Ecrire *à la main* l'algorithme des deux méthodes, puis transformez-le en programme C++.

c) Selon la méthode 2, quelle est l'influence de l'encoding du fichier sur la tokenisation du mot «tête » ?

Exercice 5 : Ecrire un programme exécutable (simple) **CompterMots** qui compte le nombre de mots d'un fichier texte (avec la méthode 1) en tenant compte de la ponctuation. Testez-le avec le fichier « litterature.txt ».

Exercice 6 : Modulariser ce programme en créant une fonction **renvoieNbreTokens** qui prend en entrée un tableau de caractères et renvoie le nombre de tokens de cette phrase.

Exercice 7 : Paramétrez cette fonction pour qu'elle puisse renvoyer le nombre de tokens selon la méthode 1 ou la méthode 2.