Travaux pratiques n° 4
Flux C++

Entrées et sorties en C++ sont gérées par des flux. Intuitivement un flux peut être vu comme un tuyau, rempli d'un côté et vidé de l'autre. L'opérateur de remplissage d'un flux se note «, l'opérateur de vidange se note ». Dans un flux peuvent être envoyées des variables de type chaîne de caractères, caractère, entier, réel, booléen [1]. Les informations peuvent être récupérées dans leur format d'origine ou bien sous forme de chaînes de caractères.

En interne, les flux traitent leur contenu comme une séquence de caractères : ce sont les opérations de remplissage et de vidange qui convertissent des suites en caractères en données du type souhaité et inversement.

1 Séparateurs

Ce fonctionnement interne implique que, dans un flux, les données significatives doivent être séparées par des caractères reconnaissables, les *séparateurs* :

— l'espace (' '),
— la tabulation ('$\backslash t$'),
— le retour à la ligne ('$\backslash n$') que vous connaissez déjà sous le nom `nl`.

Les séparateurs sont automatiquement ignorés lors des opérations de lecture de flux : la donnée lue commence au premier caractère significatif [2].

En revanche, lors des opérations d'écriture il faut prendre soin d'insérer des séparateurs afin que les données écrites dans le flux puissent être relues correctement. Par exemple, écrire les entiers 1 et 2 sans insérer entre eux un séparateur comme un espace), c'est écrire la valeur 12.

2 Flux

2.1 Flux standards : écran et clavier

Les fonctionnalités d'entrée/sortie standards en C++ sont fournies dans le fichier "iostream.h". Pour les utiliser, il faut inclure ce fichier au moyen de l'instruction #include <iostream> dans le préambule du programme.

Le flux de sortie standard s'appelle `cout` (*console output*). Le programme le remplit avec les données à afficher, le système se charge de le vider et d'afficher les données à l'écran.

Le flux d'entrée standard s'appelle `cin` (*console input*). Le système le remplit avec les données saisies au clavier, le programmr le vide pour, par exemple, affecter des valeurs à ses variables.

2.2 Flux_fichiers

Les flux_fichiers sont définis dans le fichier "fstream.h" et contenus eux aussi dans la bibliothèque standard et l'espace de noms `std`. Une variable flux_fichier de nom f est de type `fstream`. Elle se comporte comme un flux d'entrée ou de sortie selon que le fichier associé est ouvert en lecture ou en écriture.

1. vrai est codé 1, faux est codé 0.
2. Dans certaines situations, il est nécessaire de lire tous les caractères dans un flux, y compris les séparateurs. Pour ce faire il faut utiliser la fonction `getline(f,s)` qui permet de récupérer dans une variable s de type chaîne de caractères (string) la séquence de tous les caractères du flux f jusqu'au prochain retour à la ligne (exclu). Cette façon de lire de l'information non formatée d'un flux est peu compatible avec la lecture formatée au moyen de l'opérateur » : il est déconseillé de mélanger les deux usages.

2.2.1 Ouverture en lecture

Le flux_fichier f en lecture est associé à un fichier de nom n (une chaîne de caractères de type `string`) au moyen de l'appel de fonction :

```
f.open(n.c_str(),ios::in)
```

qui utilise deux paramètres :

— `n.c_str()` : en C++, l'ouverture de fichier n'accepte pas les noms en tant que `string`, c'est pourquoi il faut passer une chaîne de caractères de type `char*` (le type chaîne du langage C) en utilisant la fonction `n.c_str()` ; en revanche, le nom peut être spécifié par une chaîne de caractères constante ("...") sans problème.

— `ios::in` : ce mot clé indique le mode d'ouverture en lecture du flux_fichier.

2.2.2 Ouverture en écriture

Le flux_fichier f en écriture est associé à un fichier de nom n (une chaîne de caractères de type `string`) au moyen de de l'appel de fonction `f.open(n.c_str(),ios::out|ios::trunc)`.

Le mot clé `ios::out|ios::trunc` indique le mode d'ouverture en écriture du flux_fichier (il existe d'autres modes d'ouverture en écriture en C++ mais nous ne les traiterons pas cette année).

2.2.3 Contrôle de l'ouverture

En C++, il est possible de vérifier que l'ouverture a bien été effectuée au moyen de la fonction `f.is_open()` qui retourne vrai ssi c'est le cas.

2.2.4 Lecture

Après ouverture, le flux peut être utilisé comme le flux d'entrée standard au moyen de l'opérations » [3].

2.2.5 Écriture

Après ouverture, le flux peut être utilisé comme le flux de sortie standard au moyen de l'opération «.

2.2.6 Détection de la fin de fichier lors de la lecture

La fonction `f.eof()` (pour End Of File) indique si la tête de lecture a atteint la fin du fichier.

De façon plus générale, les problèmes de lecture de fichier sur ordinateur pouvant être nombreux, il est possible d'utiliser la fonction `f.good()` qui retourne vrai ssi la lecture est possible dans le flux_fichier f [4].

2.2.7 Fermeture

En fin de traitement (lecture ou écriture), l'opération `f.close()` libère la tête de lecture/écriture.

3. et `getline` si nécessaire.
4. `f.good()` subsume `f.eof()`.