

Cours de C++, 1ère année, HE-Arc

Série 9.1

Exercice 1 : Ecriture binaire

Ecrire un programme qui enregistre (sous forme binaire, et non pas formatée), dans un fichier dont le nom est fourni par l'utilisateur, une suite de nombres entiers fournis sur l'entrée standard. On conviendra que l'utilisateur fournira la valeur 0 (qui ne sera pas enregistrée dans le fichier) pour préciser qu'il n'a plus d'entiers à entrer.

Dans un deuxième temps, on fera les tests suivants :

- Si le fichier existe déjà, on fera une demande de confirmation.
- Dans la lecture des entiers, on veillera à surveiller que l'utilisateur n'entre que des entiers.

Exercice 2 : Lecture binaire

Ecrire un programme permettant de lister (sur la sortie standard) les entiers contenus dans le fichier tel que celui créé par l'exercice précédent.

Exercice 3 : Rang

Ecrire un programme permettant à un utilisateur de retrouver, dans un fichier tel que celui créé dans l'exercice 1, les entiers dont il fournit le « rang » (sa valeur). On conviendra qu'un rang égal à 0 signifie que l'utilisateur souhaite mettre fin au programme.

Exercice 4 : Lecture formatée

Ecrire un programme qui ouvre un fichier texte dont le nom est donné par l'utilisateur et qui affiche le contenu sur la console en numérotant les lignes.

REMARQUE/RAPPEL

La méthode `ignore` vide le contenu pouvant être dans le flux du nombre de caractères `n` spécifié, soit jusqu'à ce que le caractère spécifié `delim` soit rencontré. C'est un peu l'équivalent de `fflush(stdin)` pour le C.

Syntaxe :

```
istream& ignore( int n=1, int delim=eof() );
```

Par exemple :

```
cin.ignore( numeric_limits<streamsize>::max(), '\n' );
```

Pour les autres méthodes utiles* pour la lecture/écriture avec un fichier en C++, se référer p.ex. aux documents supplémentaires 19_Fichiers.pdf et 19b_Fichiers.pdf.

- `get`, `getline`, `ignore`, `open`, `close`, `fail`, `clear`, `write`, `read`, `seekg`, ...