

La gestion des entrées-sorties proposée par la bibliothèque *iostream* en C++ sont assez lentes. Jusqu'à présent, vous ne l'avez pas forcément remarqué car les exercices ne demandaient pas de lire ou d'écrire beaucoup de choses. À partir de maintenant, cela pourra être le cas ; aussi voici quelques conseils et éléments de syntaxe pour accélérer les entrées/sorties en C++.

Lecture rapide

Pour une grande valeur de *nbValeurs*, on considère le pseudo-code suivant et on suppose que le programme C correspondant met **1 seconde** à s'exécuter :

```
Répéter nbValeurs fois  
  Lire un entier
```

Alors un programme C++ basé sur le code suivant mettra presque **3 secondes** à s'exécuter :

```
for (int ival = 1; ival <= nbValeurs; ival++)  
{  
    cin >> valeur;  
}
```

Pour le rendre bien plus rapide, il suffit d'ajouter la ligne suivante juste après le début de la fonction *main* :

```
ios::sync_with_stdio(false);
```

Le code obtenu ne mettra plus qu'environ **1,1 secondes** à s'exécuter, soit presque la même vitesse qu'un programme C.

Explication technique : C++ permet d'utiliser en parallèle les entrées C++ et les entrées C. Les deux doivent donc être synchronisés ; en effet, les entrées-sorties utilisent un espace mémoire appelé « tampon » qui récupère plein de données d'un coup pour diminuer le nombre d'opérations effectuées avec le flux, qui sont plus lentes que des opérations en mémoire. La synchronisation prend ainsi un certain temps ; en la désactivant, on peut donc nettement accélérer l'exécution.

Écriture rapide

Pour une grande valeur de *nbValeurs*, on considère le pseudo-code suivant et on suppose que le programme C correspondant met **1 seconde** à s'exécuter :

```
Répéter nbValeurs fois  
  Afficher un entier puis un retour à la ligne
```

Alors un programme C++ basé sur le code suivant mettra presque **40 secondes** à s'exécuter :

```
for (int ival = 1; ival <= nbValeurs; ival++)  
{  
    cout << valeur << endl;  
}
```

Pour le rendre bien plus rapide, il suffit de remplacer `endl` par `'\n'` :

```
for (int ival = 1; ival <= nbValeurs; ival++)  
{  
    cout << valeur << '\n';  
}
```

Le code obtenu ne mettra plus que environ **1,3 secondes** à s'exécuter soit seulement 30% plus lent qu'un programme C.

Explication technique : Il s'agit à nouveau d'une question de tampons : quand on écrit sur la sortie, le texte ne s'affiche pas tout de suite, car C++ le stocke et attend d'avoir suffisamment de caractères pour les afficher, pour réduire le nombre d'opérations. Le manipulateur *endl* permet d'effectuer l'affichage tout de suite, car sur une interface utilisateur, c'est nécessaire quand on revient à la ligne. Mais pour nos exercices, c'est inutile : il vaut mieux profiter du tampon. On gagne donc beaucoup de temps en utilisant un simple caractère de retour à la ligne : `\n`.

Utiliser les entrées-sorties du C

Relativement faciles à apprendre, elles sont naturellement plus rapides et permettent de personnaliser l'affichage plus simplement. Un cours dédié sera bientôt mis en ligne.