

Corrigé du TP3

Disclaimer : Ce corrigé n'a pas pour but de vous fournir un détail des algorithmes répondant aux questions de l'énoncé, mais de vous résumer les notions clés dont vous aurez besoin pour résoudre ces exercices, les avertissements vis à vis de certaines erreurs classiques et les détails des passages les plus techniques, voire des précisions si vous souhaitez aller plus loin sur ces questions et plus généralement de synthétiser dans un même document la plupart des commentaires fait en TP afin qu'ils puissent profiter à tous. Pour obtenir un corrigé détaillé, il vous faudra donc vous référer à ce que vous avez pu faire en TP ou reprendre vous même les exercices à partir des notions présentées dans le présent corrigé pour écrire les algorithmes qui vous manquent.

En cas de difficultés, **n'hésitez pas à me contacter**, nous pourrons en discuter soit par mail soit en début de séance suivante. Ce corrigé n'a pour but que d'être une base de travail pour vous et de servir de complément par rapport à ce que vous avez fait en séance, il n'est en aucun cas fait pour se suffire à lui même.

Manipuler une chaîne de caractère

- * Le type d'une chaîne de caractère est **string**.
- * Une chaîne de caractère n'est rien d'autre qu'un tableau de caractère, elle se manipule d'une façon très similaire aux vecteurs.
- * Vous pouvez connaître la taille d'une chaîne de caractère avec la commande **.size()**
- * Vous pouvez accéder au caractère en position n avec la commande **chaîne[n]**
Les chaînes sont numérotées à partir de 0 et non 1 comme les vecteurs.
- * Vous pouvez ajouter un caractère c à la fin d'une chaîne avec la commande **chaîne__pushback(c)**
- * Vous pouvez également concaténer deux chaînes avec un +

Lire dans un fichier

- * Pour lire dans un fichier, vous aurez besoin de la bibliothèque **fstream**.
 - * Vous devez au préalable l'ouvrir avec la commande : **ifstream nomDuFlux("chemin")**
Si vous indiquez un chemin absolu dans la commande ifstream, chaque dossier devra être séparé par un slash et non un backslash. Un copier/coller du chemin pour certain ordinateur ne vous donnera pas les bons séparateurs.
Si vous indiquez un chemin relatif, il est important que votre fichier soit **dans le même dossier** que votre projet en .cbp. Vous ne pouvez pas utiliser de chemin relatif sous Mac OS.
 - * Une fois le fichier ouvert pour la lecture, il existe trois façons de le lire
 - Caractère par caractère avec la commande :
char c ; nomDuFlux.get(c) ;
 - Mot par mot avec la commande :
string mot ; nomDuFlux >> mot ;
 - Ligne par ligne avec la commande :
string ligne ; getline(nomDuFlux,ligne) ;
- À l'ouverture du fichier, un curseur est placé au début. Lorsqu'un caractère, mot ou ligne est lue, le curseur passe automatiquement au prochain caractère, mot ou ligne.
- * Pour lire un fichier, on suit la procédure suivante :
 1. On ouvre le fichier, le curseur est alors placé au début.
 2. On lit le premier élément (caractère, mot ou ligne en fonction de ce que l'on choisit), on le stocke dans une variable, le curseur passe au suivant.
 3. On lit tous les éléments les uns après les autres en utilisant une boucle **while** jusqu'à arriver à la fin

du fichier.

4. Pour repérer la fin du fichier, il y a deux façons de faire :
 - On peut utiliser la commande `nomDuFlux.eof()`, pour end of file, qui renvoie *true* si le curseur est à la fin du fichier et *false* sinon. La condition de la boucle `while` est alors `!nomDuFlux.eof()`
 - On peut également utiliser la commande `getline(nomDuFlux,ligne)` qui affecte la ligne suivant le curseur à la variable `ligne` et qui renvoie *false* si l'on est à la fin du fichier et *true* sinon. La condition de la boucle `while` est alors `getline(nomDuFlux,ligne)`

ATTENTION. Les commandes `getline` et `eof` renvoient l'inverse l'une de l'autre.

- * N'oubliez pas de fermer les flux une fois que vous en avez terminé avec vos fichiers par la commande `nomDuFlux.close()` Cela vous évitera des erreurs dans des programmes plus complexes.

Écrire dans un fichier

- * Pour écrire dans un fichier, vous aurez besoin de la bibliothèque `fstream`.
- * Vous devrez au préalable l'ouvrir, cette fois avec la commande : `ofstream nomDuFlux("chemin")`
- * Attention, la commande `ofstream` ne prend que des char et non des strings, il vous faudra donc convertir votre nom de fichier avec la commande `nomDeFichier.c_str()`
- * Pour écrire dans un fichier, c'est exactement comme pour utiliser `cout`, mais cette fois, ce n'est pas vers le terminal que vous l'envoyez, mais vers le fichier, utilisez donc au lieu de `cout` le nom du flux défini avec `ofstream`, par exemple : `nomDuFlux << "Hello world" << endl`
- * Écrire dans un fichier qui existe déjà le supprimera.

Si vous souhaitez écrire à la suite d'un fichier existant déjà, il faudra utiliser la commande : `ofstream nomDuFlux("chemin",ios::app);`

- * N'oubliez pas de fermer les flux une fois que vous en avez terminé avec vos fichiers par la commande `nomDuFlux.close()` Cela vous évitera des erreurs dans des programmes plus complexes.

Quelques conseils pour retravailler le TP

- * Il est nécessaire que vous intégrez que les chaînes de caractères s'utilisent et se manipulent comme des vecteurs. À partir de là, la difficulté revient à savoir manipuler correctement des vecteurs.
- * La lecture et l'écriture dans des fichiers peut sembler lourde, mais il n'y a que peu de commandes à connaître. Nous en ferons encore un peu par la suite, mais cela ne reviendra que rarement, c'est pourtant primordial dès que vous souhaitez faire du traitement de données. Notamment traiter des résultats de simulations numériques ou des données expérimentales. Il est important pour vous d'être à l'aise avec ces notions.