
Q: Est-ce qu'on doit ouvrir un fichier?

R: Non. Vous n'avez pas besoin d'ouvrir un fichier.

Dans le « Tp#1 », vous avez redirigé la sortie vers un fichier « xyz.txt » en utilisant l'opérateur « > ». Est-ce que vous avez ouvert le fichier « xyz.txt » pour écrire les résultats obtenus? Non. C'est le système qui en voyant l'opérateur « > » a ouvert le fichier pour vous et a écrit les données pour vous.

Idem pour l'opérateur de « < ». Si vous décidez de l'utiliser, le système va ouvrir pour vous le fichier et va lire pour vous son contenu.

Q: Comment lire alors les données?

R: En utilisant le cin!

Q: Comment?

R: C'est juste une lecture. Il vous manque juste à faire une petite analyse très simple pour savoir quoi utiliser avec le « cin ». Il y a plusieurs façons de le faire. Se servir des informations fournies en annexe -1- et/ou -2- ou tout simplement les opérateurs de structures de contrôle.

Q: Je ne vois pas le lien entre l'annexe -1- et le reste de l'énoncé, pourrais-je avoir des explications ?

R: D'après moi, elle n'est pas utile. Mais comme chacun a sa propre façon de coder, je voulais anticiper une question pour ceux/celle qui veulent savoir comment avoir le nombre d'arguments sur la ligne de commande. Idem pour l'annexe 2.

Q: C'est quoi la différence entre:

tp2.exe M + M

(et)

tp2.exe < inx.txt

R: Elles sont identiques!

-1- que vous fassiez: tp2.exe M + M

-2- ou bien que vous écriviez dans par exemple le fichier "inx.txt" l'expression: $M + M$ puis vous lancez le programme avec la commande: `tp2.exe < inx.txt`

Techniquement parlant -1- et -2- correspondent à la même opération. La 2e opération, elle évite de taper à chaque fois l'opération $M + M$!

Vous pouvez vous entraîner avec le petit exercice suivant (fichier test.cpp):

```
include <iostream>
using namespace std;
int main() {
    cin >> x;
    cout << x;
    return 0;
}
```

Vous exécutez le programme avec comme argument 1:

`test.exe 1`

Puis vous créez le fichier fic.txt et vous écrivez à l'intérieur le chiffre 1 et vous exécutez la commande suivante:

`test.exe < fic.txt`

Ça revient à la même chose!

Q: Dans votre énoncé de tp02, il est indiqué, dans le barème, que 3.5point des notes est accordés pour les tests. De quel test s'agit-il?

R: Les tests fournis: les fichiers in/out 1 à 4. Les tests non fournis: d'autres tests avec d'autres opérations mathématiques mais tout en respectant l'énoncé. On ne va pas tester par exemple la multiplication!
