

- **Programmer le TD2: (Exercice de la Série TD2)**

Reprendre tous les exercices de la Série du TD 2. Programmer en C++.

- **Exercices sur les fichiers: (Encore des fichiers)**

Exercice_1: Utiliser l'instruction **fputc** pour écrire un programme qui réécrit son code dans un autre fichier. Par exemple si ce programme s'appelle `exo_n.c`, alors il crée un fichier `copie_exo_n.c` qui est une copie de lui-même.

Exercice_2: Plusieurs fonctions permettent de lire et écrire dans des fichiers. Il existe entre autres **fread** et **fwrite** qui s'emploient comme suit:

```
int fread(void * tableau, size_t nb_octets, size_t n, FILE *f).
```

void * *tableau*: est un tableau dont le contenu peut être des caractères ou toute autre chose (d'où, on utilise le mot **void**). Chaque caractère est représenté par un octet. Exemple: *tableau* peut être déclaré comme suit **char**tableau***.

size_t: un nom de type prédéfini en C++ qui signifie un type dont les éléments peuvent être des valeurs très grandes (représentées sous formes de plusieurs octets).

Cette fonction tente de lire, du fichier **f**, **n** blocs de taille **nb_octet** octets, et les écrit dans **tableau**, qui doit donc être un tableau suffisamment grand que l'on aura créé au préalable. **fread** renvoie le nombre de blocs lus, qui peut être plus petit que **n**, par exemple si l'on a atteint la fin du fichier, et elle renvoie 0 en cas d'erreur.

Remarque: le premier paramètre effectif lors de l'appel de **fread** peut être tout simple **&c** tel que **c** est un caractère. On rappelle que **&c** donne l'adresse de **c** (ceci est présenté dans la séance de TD).

```
int fwrite(void *tableau, int taille, int n, FILE *f)
```

agit de manière similaire: elle écrit, dans le fichier **f** un nombre **n** de blocs de taille **taille** octets qu'elle lit depuis **tableau**. Elle renvoie le nombre de blocs écrits en résultat. Si il est inférieur à **n**, c'est qu'il y a eu une erreur.

Questions:

- 1) Utiliser la fonction **fread** pour programmer la fonction **fgetc**, que nous avons déjà vu dans TD et TP;
- 2) Utiliser la fonction **fwrite** pour programmer la fonction **fputc**, que nous avons déjà vu dans TD et TP;

Exercice_1: Faites un programme qui prend en argument sur la ligne de commande le nom d'un fichier, et une lettre (c, m ou l) et qui renvoie selon cette lettre soit le nombre de caractère, de mots ou de lignes du fichier.

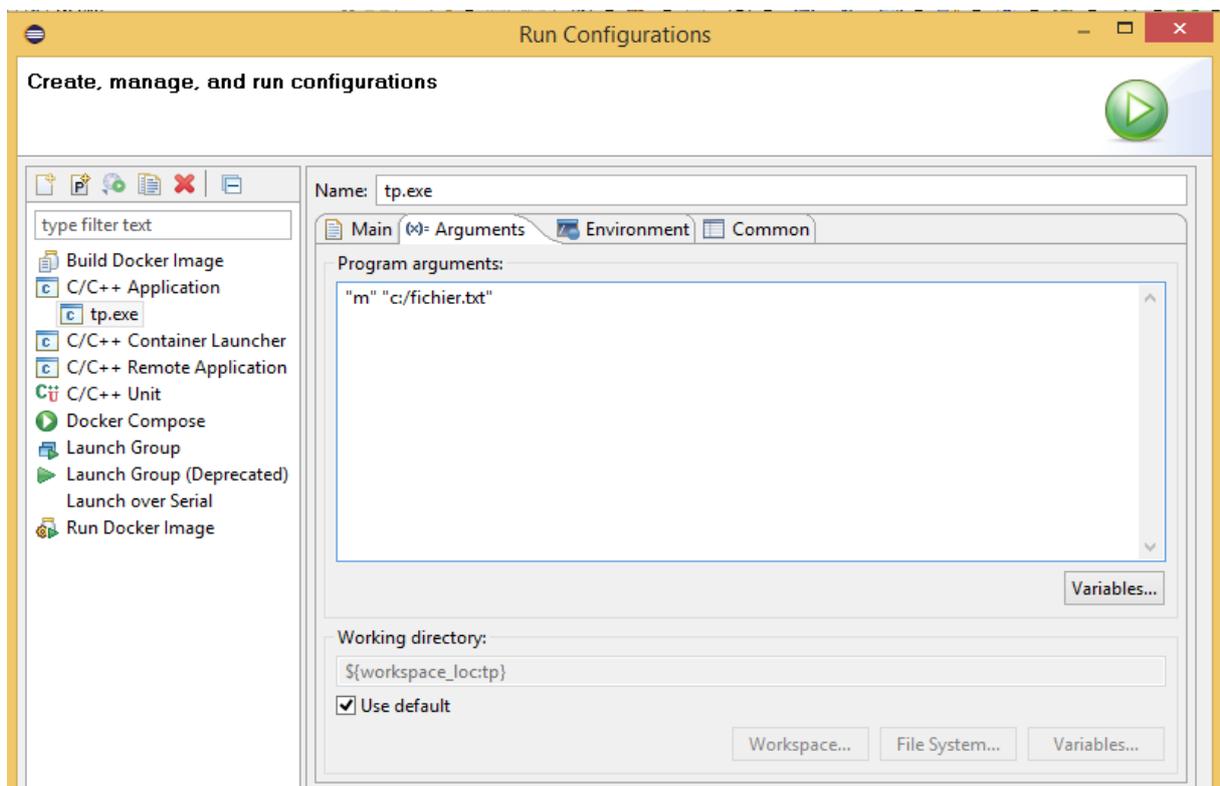
Pour faire ce programme, il faut travailler le main ayant des arguments comme suit:

```
int main(int argc, char** argv)
```

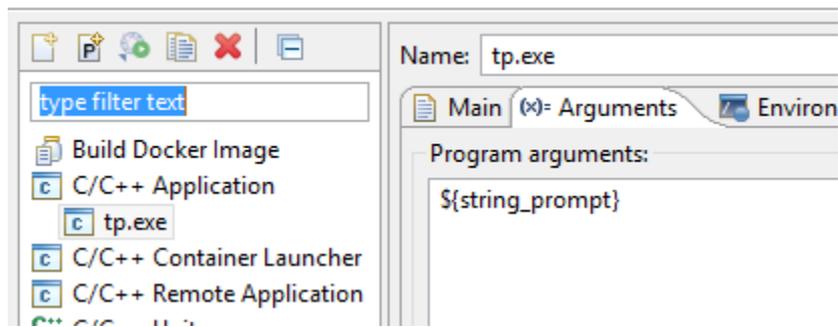
Ceci veut dire que le main a deux paramètres:

- **char** argv:** argv est un tableau de chaînes de caractères. Il contient les paramètres passés au programme main;
- **int argc:** argc est un entier qui donne le nombre de paramètres passés + 1 (Le paramètre argv[0] contient toujours nom de l'exécutable : le nom de ton programme tp.exe par exemple).

- 1) Pour exécuter ce programme, il faut configurer Eclipse pour qu'il prenne en considération les paramètres du programme main ; Pour faire ceci, vous pouvez procéder comme suit. Mettre les paramètres à passer sous formes de chaînes de caractères séparées par espace dans Run->Run Configurations->Arguments. N'oublier pas Apply et run:

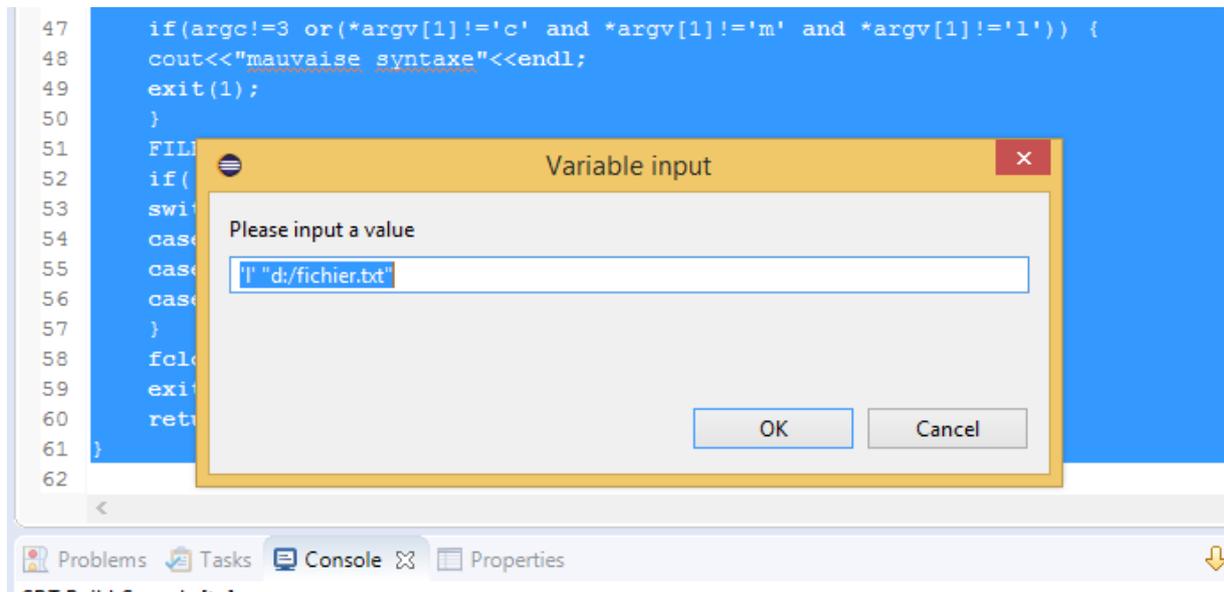


On peut même passer ces paramètres dans une fenêtre si on met dans l'onglet Argument de Eclipse le texte : `$(string_prompt)`, comme suit :



Et lors de l'exécution du programme, vous allez avoir :

Pour saisir les paramètres : argv[1] et argv[2] sous formes de deux données, exemple : `l` et `"d:/fichier.txt"`.



- 2) Maintenant, dans votre programme **main** et avant de commencer la lecture du fichier, il faut commencer par vérifier que le nombre **argc** est égal à 3 (2 arguments+1). Les deux arguments argv[1] et argv[2] sont le **nom du fichier** est l'un **des caractères** : c, m, ou l) ;