

# TP à rendre 1

Dans cet exercice vous devez écrire le programme `compare` qui reproduit le comportement de la commande Unix `cmp` (mais sans les options). La syntaxe est la suivante :

```
./compare <fichier1> <fichier2>
```

Ce programme `compare`, octet par octet, le contenu des fichiers `fichier1` et `fichier2`. Votre programme doit reproduire exactement le comportement suivant : si les fichiers sont identiques, le programme termine avec un code de retour nul et les sorties standard et erreur sont vides ; dans le cas contraire, le programme termine avec un code de retour non nul et affiche un message spécifique sur la sortie erreur. Ces messages sont :

- lorsqu'un octet est différent :  
« `fichier1 fichier2 differ: byte x, line y` »
- lorsque la comparaison atteint la fin de l'un des fichiers alors que l'autre comporte encore des octets :  
« `EOF on fichier1 after byte x, line y` »
- lorsqu'un fichier est vide et pas l'autre :  
« `EOF on fichier1 which is empty` »

Vous écrirez le programme `compare` en langage C en utilisant uniquement les primitives système. Les seules fonctions de bibliothèque autorisées sont pour l'affichage (par exemple `fprintf` ou `perror`). L'usage d'une primitive système étant coûteux en termes de ressources, vous veillerez à les utiliser avec parcimonie. Votre programme doit vérifier les valeurs de retour de toutes les primitives système utilisées et ne devra pas comporter d'allocation dynamique de mémoire.

Vous apporterez un soin particulier à la mise en forme de façon à rendre un code lisible et commenté à bon escient. Référez-vous au document « Conseils de programmation pour réussir vos projets et TP » (mis à votre disposition sur Moodle) et, si besoin, utilisez l'utilitaire `clang-format` avec la configuration donnée dans ce document.

Votre programme doit compiler avec la commande suivante (disponible dans le `Makefile` mis à disposition sur Moodle) : `cc -Wall -Wextra -Werror`

Les programmes qui ne compilent pas avec cette commande **ne seront pas examinés**. Un script de test est mis à votre disposition sur Moodle. Celui-ci exécute votre programme sur des jeux de tests qui serviront de base à l'évaluation de votre rendu. La commande suivante permet de lancer les tests :

```
make test
```

N'hésitez pas à contacter votre enseignant si vous constatez un comportement anormal ou si vous souhaitez ajouter un test.

Vous devrez rendre sur Moodle un *unique* fichier nommé `compare.c` (Moodle sait qui vous êtes, il est inutile d'appeler votre programme `Jean-Claude_Dusse_compare.c`, et il est interdit de rendre un fichier d'un autre nom ou une archive au format du jour). De plus, assurez-vous de rendre des fichiers utilisant l'encodage UTF-8 — il y aura de sévères pénalités sinon.

Ce TP à rendre est **individuel**. On rappelle que la copie ou le plagiat sont sévèrement sanctionnés.