

Exercice 1

Difficulté : 42 points-virgules

On souhaite rédiger un programme `cprep` pour recopier plusieurs fichiers dans un répertoire donné, en utilisant des lectures/écritures de taille variable. Le programme doit admettre la syntaxe suivante :

```
cprep taille repertoire f1 ... fn
```

Le répertoire de destination est créé s'il n'existe pas déjà, puis les fichiers f_1 à f_n sont recopiés dans ce répertoire en utilisant des lectures et des écritures de *taille* octets à la fois. Les permissions et les dates des fichiers doivent également être recopiées. Le nombre de fichiers fournis sur la ligne de commande doit être supérieur ou égal à 1, mais ne doit pas être limité.

Par exemple :

```
> ./cprep 1024 /tmp/toto /usr/include/stdio.h cprep.c
> ls -l /tmp/toto
total 36
-rw-r--r-- 1 pda users 2111 Feb  3 13:37 exo1.c
-rw-r--r-- 1 pda users 31526 Oct  7 10:13 stdio.h
```

Pour rédiger votre programme, il est impératif de respecter les contraintes suivantes :

- vous ne devez utiliser que les primitives système (ou assimilées comme telles); vous pouvez toutefois utiliser les fonctions de bibliothèque pour les affichages ou les manipulations de chaînes de caractères;
- pour des raisons d'efficacité, vous ne ferez pas d'appels redondants à des fonctions lentes (primitives système ou autres);
- pour des raisons de simplicité, vous limiterez la taille du chemin des fichiers créés à la constante `CHEMIN_MAX` que vous définirez à 512 octets : un chemin plus long doit être considéré comme une erreur;
- vous vérifierez soigneusement les débordements de tableau (vous pouvez notamment utiliser la fonction de bibliothèque `snprintf` pour contrôler la taille de chaînes complexes);
- votre programme doit retourner un code de retour nul (`exit(0)`) si tout s'est déroulé sans erreur ou un code de retour non nul (`exit(1)`) si une erreur a été rencontrée;
- si votre programme est appliqué avec un nombre d'arguments incorrect, il doit afficher un message de la forme : `"usage : cprep taille rep f1 ... fn"`.
- 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 pour réussir vos TP et projets » mis à votre disposition sur Moodle et, si besoin, utilisez l'utilitaire `clang-format` avec la configuration donnée dans ce document;
- vous trouverez sur Moodle une archive contenant des exemples et divers fichiers pour vous aider;
- votre programme doit compiler avec les options `-Wall -Wextra -Werror` sur `gcc` version 9.4 minimum (la version disponible sur la machine `turing.u-strasbg.fr`). Alternativement, vous pouvez utiliser l'image Docker `pdagog/refc` (version de `gcc` 12.2) Les programmes qui ne compilent pas au moins sur `turing` avec ces spécifications **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 : `sh test1.sh`.

Vous devrez rendre sur Moodle un *unique* fichier nommé `cprep.c`.

Cet exercice est **individuel**. On rappelle que la copie ou le plagiat sont sévèrement sanctionnés.