



Travaux Pratiques d'algorithmique n°6

Cours d'Informatique de Deuxième Année

—L2.1 Sciences—

Manipulation de fichiers

Pour manipuler des fichiers en C, il faut utiliser une variable de type `FILE *`. Les fonctions suivantes permettent la lecture/écriture dans un fichier (**man fonction** pour plus d'informations sur **fonction**):

- `FILE *fopen(char *nom, char *mode);` : ouvre le fichier dont le nom est le premier paramètre selon le mode spécifié par le deuxième paramètre. Les principaux modes sont : "r" pour la lecture, "w" pour écriture et "a" pour l'ajout. Cette fonction retourne `NULL` en cas d'erreur;
- `int fprintf(FILE *f, char *format,...);` et `int fscanf(FILE *f, char *format,...);` : ont le même comportement que les fonctions `printf` et `scanf`, excepté le fait qu'elles écrivent/lisent dans le flux passé en premier argument. En particulier, `fscanf` retourne la constante particulière `EOF` (de type `int`) lorsque la fin du fichier est atteinte;
- `int fclose(FILE *f);` : ferme le flux `f`.

Vous prendrez soin de bien tester les valeurs de retour de ces fonctions et de fermer les fichiers que vous avez ouvert. Les éventuelles erreurs ne devront pas entraîner l'arrêt du programme.

► Exercice 1. Echo

Ecrire un programme (portant le nom `Monecho`), qui affiche sur la sortie standard les paramètres qu'il a reç en argument de la ligne de commande.

Exemple d'utilisation :

```
$> ./Monecho bonjour toto
bonjour toto
$>
```

► Exercice 2. Catalogue

Écrire un programme `catalogue` qui effectue l’affichage du contenu d’un fichier sur la sortie standard. Le nom du fichier à traiter sera transmis au programme comme argument de la ligne de commande. L’exécutable sera donc lancé à partir du shell sous forme :

```
>$ ./catalogue f.
```

Transformer le programme précédent pour qu’il puisse traiter une suite de fichiers. Le lancement se fera sous forme :

```
>$ ./catalogue f1 f2 ...fn.
```

► Exercice 3. Traitements

On souhaite maintenant effectuer des traitements sur le contenu des fichiers dont les noms sont transmis en arguments de la ligne de commande. Le type de traitement à effectuer sera ajouté en premier argument du programme précédent (sans type de traitement, le programme devra se comporter comme dans l’exercice précédent).

- **Recopie.** Si le premier argument du programme est “-b”, le programme devra alors écrire dans un fichier (portant le même nom que le fichier d’entrée mais suffixé par l’extension “.backup”) l’ensemble du contenu du fichier d’entrée. L’allocation de la chaîne de caractères recevant le nom du fichier devra être effectué dynamiquement.

Exemple d’utilisation :

```
$> ./catalogue -b f1 f2
```

Recopie l’ensemble du fichier `f1` dans un fichier `f1.backup` et l’ensemble du fichier `f2` dans un fichier `f2.backup`.

- **Mise en évidence des mots.** Si le premier argument du programme est “-w”, le programme devra alors écrire dans un fichier (portant le même nom que le fichier d’entrée mais suffixé par l’extension “.words”) l’ensemble des mots du fichier d’entrée.
- **Mots en majuscules.** Si le premier argument du programme est “-uc”, le programme effectuera le même traitement qu’avec l’option “-w”, mais vous prendrez soin de transformer les mots en majuscule lors de l’écriture dans le fichier (suffixé par “.uppercase”).
- **Mise en évidence des entiers.** Si le premier argument du programme est “-i”, le programme devra alors écrire dans un fichier (portant le même nom que le fichier d’entrée mais suffixé par l’extension “.integers”) l’ensemble des entiers du fichier d’entrée.