TP 7 : LECTURE/ÉCRITURE DANS DES FICHIERS

Table des matières

But

Vous devez maitriser à la fin de cette séance la manipulation des fichiers en lecture et en écriture.

Exercice 1 : mycat

Écrire une fonction `int mycat(char *fichier_source1, char* fichier_source2, char *fichier_out)` qui met bout à bout les `fichier_source1` et `fichier_source2` dans le `fichier_out`. La fonction renvoie 0 si une erreur se produit et 1 si la concaténation s'est bien passée.

Exercice 2 : lecture formatée

On dispose d'un fichier `constantes.txt` contenant les lignes suivantes :

```
pi 3.14159265358979323846264338327950288419716939937510582O
nombre_d_or 1.6180339887498948482045868343656381177203091798O5
e 2.7182818284590452353602874713526624977572470936999595749669676277240766303535
```

Écrire un programme qui lit ce fichier, et affiche à l'écran un tableau bien normalisé où chaque constante est exprimée avec six chiffres après la virgule :

```
pi 3.141593
nombre_d_or 1.618034
e 2.718282
```

Exercice 3 : fgetc

Écrire une fonction `int count_char(char *fichier, char carac)` qui calcule le nombre d'occurrences du caractère `carac` dans le fichier `fichier`.

Exercice 4 : lecture/écriture formatée

Écrire un programme constitué de deux fonctions :

(1) La fonction `void ajout(char *chemin, int n)` permet de saisir `n` noms de personnes au clavier et de stocker ces noms, en les séparant par des tabulations ($\widehat{()}$), dans le fichier de nom `chemin` ;

(2) La fonction `void afficher(char *chemin, int n)` permet de lire `n` noms de personnes à partir du fichier de nom `chemin`.