

Exercices prog. C – Fichiers + corr

Ecrire le programme **Reverse** qui va sauver dans un fichier de sortie les lignes lues dans un fichier d'entrée, mais avec chaque ligne du fichier de sorties sont écrites à l'envers.

Ex.

12345 6789 abcd

devient

dcba 9876 54321

Attention à la gestion des retours à la ligne.
Gestion des erreurs fichier au « plus simple ».

Hypothèses :

Une ligne du fichier d'entrée ne fait pas plus de 255 caractères.

Le caractère de retour de ligne est '\n'.

```
#include <stdio.h>
#include <string.h>
#define size 256

int main(void) {
    FILE *f_in, *f_out;
    char tmp[size], reverse[size];
    f_in = fopen("123.txt", "r");
    f_out = fopen("321.txt", "w");

    if(f_in == NULL){ perror("Erreur lors de l'ouverture \n"); return -1; }
    if(f_out == NULL){ perror("Erreur lors de l'ouverture\n"); return -1; }

    while (fgets(tmp, size, f_in) != NULL) { // fgets() ligne à ligne ou max size
        int HasReturn = (tmp[strlen(tmp)-1]) == '\n'; // last char is '\n' ?
        if (HasReturn) { // remove '\n'
            tmp[strlen(tmp)-1] = '\0';
        }
        for(int i = 0; i < strlen(tmp); i++) // copy in reverse order chars
            // from tmp to reverse
            reverse[i] = tmp[strlen(tmp)-i-1];

        if (HasReturn) // add '\n' at the end of reverse
            reverse[strlen(tmp)] = '\n';

        reverse[strlen(tmp)+1] = '\0'; // update reverse eol, might be
            // garbage

        fprintf(f_out, "%s", reverse);
    }
    fclose(f_in);
    fclose(f_out);
    return 0;
}
```