

THE



C



CAVE

A place of basic low-level programming

Tutoriel 4 - Opérateurs sur les bits et commandes système.

Vous en êtes-vous sortis pour résoudre le problème du mois dernier ? Comparez avec la solution que voici :

Solution du défi

```
#include <stdio.h>
#include <stdlib.h>
int newMask() {
    int mask = (double)rand()/RAND_MAX*254+1;
    return mask;
}

int main(int argc, char *argv[]) {
    int seed = 0xA3, mask = 0;
    char c;
    FILE *inputFile = 0, *outputFile = 0;

    srand(seed); /* Fixe la valeur de la graine. */

    /* Vérifie le nombre d'arguments */
    if(argc!=3) {
        printf(" Utilisation : %s <fichier source> <fichier destination>\n",argv[0]);
        return 1; /* Signale une erreur */
    }

    inputFile = fopen(argv[1],"r"); /* Ouvre le fichier source. */
    if(!inputFile) return 2;

    outputFile = fopen(argv[2],"w"); /* Ouvre le fichier destination. */
    if(!outputFile) return 3;

    c = fgetc(inputFile); /* Lit le premier caractère. */

    /* Boucle jusqu'à ce que la fin du fichier soit atteinte. */
    while(c != EOF) {
        mask = newMask(); /* Récupère une valeur pour le nouveau masque. */
        printf("mask = %d\n",mask);
        c ^= mask; /* OU exclusif avec le masque. */
        fputc(c,outputFile); /* Écriture dans le fichier destination. */
        c = fgetc(inputFile); /* Lit un autre caractère. */
    }

    /* Ferme les fichiers. */
    fclose(inputFile);
    fclose(outputFile);

    return 0;
}
```

La solution utilise un nouveau masque pour chiffrer chaque caractère. Les nombres renvoyés suivent une série, qui est répétée pour une valeur donnée de la graine en entrée. De plus, la clé de chiffrement est la graine aléatoire.