

Codes correcteurs

Exercice 1 :

Ecrire une fonction `int bruite(int octet, float p, int i)` qui prend en paramètre un `unsigned char` et qui lui applique un bruit d'intensité `i` selon la probabilité `p` avant de retourner l'octet bruité.

Le paramètre `p` est un réel qui indique la probabilité d'apparition de bruit pour l'octet passé en paramètre. Il est compris entre `0.0` et `1.0` tel que pour `0.0` il n'y a aucune chance que l'octet soit bruité et pour `1.0`, l'octet est bruité de façon certaine.

Le paramètre `i` indique l'intensité du bruit. Il s'agit du nombre de bits à modifier dans l'octet si celui-ci est bruité. Part exemple, si l'intensité est de 2, alors 2 bits doivent être modifiés dans l'octet passé en paramètre.

Par exemple, soit l'octet de valeur initiale `10110100`. L'appel à la fonction `bruite(octet, 0.5, 2)` retournera une fois sur deux un octet dont 2 bits sont modifiés (par exemple `11100100`).

Vous utiliserez les opérations de modification de variables bits à bit pour appliquer le bruit.

Exercice 2 :

Ecrire une fonction `void copie(const char* input_file, const char* output_file)` qui copie le contenu du fichier pointé par `input_file` vers le fichier `output_file`. Le fichier d'entrée sera lu caractère par caractère en utilisant la fonction `int fgetc(FILE*)` et le fichier de sortie sera écrit en utilisant la fonction `int fputc(int, FILE*)`. Vous trouverez de la documentation sur ces fonctions à l'adresse <https://openclassrooms.com/fr/courses/19980-apprenez-a-programmer-en-c/16421-manipulez-des-fichiers-a-laide-de-fonctions>. Entre chaque lecture / écriture, un bruit sera appliqué au caractère lu en utilisant la fonction `bruite` de l'exercice 1.

Ecrire une fonction `main` permettant de tester votre copie bruitée de fichier. Vous testerez les cas suivants :

Cas	Proba de bruit	Intensité
1	0.25	1
2	0.5	2
3	0.75	3