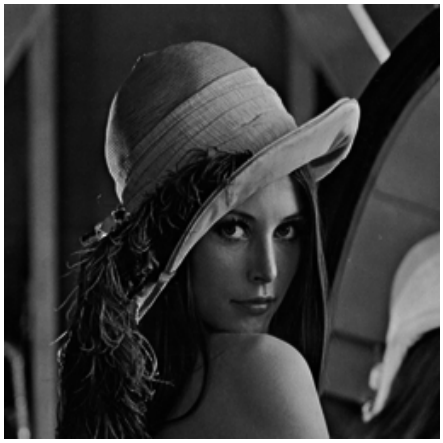
Code de Hamming et images au format PGM

L'objectif de ce TP est de mettre en application le code de Hamming (7,4,3) afin de protéger la transmission d'images au format PGM via un canal où pour chaque octet émis, au plus une erreur peut se produire.

La structure d'un fichier au format PGM (Portable Gray Map) est la suivante (cf. fig. 1) :

- une ligne contient sur 2 octets le « nombre magique » du format. Ce dernier permet de spécifier si l'image est en niveau de gris ou en couleur et si les pixels de l'image sont codés au format binaire (suite d'octets) ou au format texte ascii (suite de caractères numériques représentant la valeur des pixels). Pour ce TP, on ne traitera que le cas du nombre magique **P2** qui correspond à des images en niveaux de gris, chaque pixel étant encodé par une chaîne de caractères numériques qui est un entier entre 0 et le niveau maximum de gris utilisé (moins un). Pour ce TP, on travaillera sur des images codées en 256 niveaux de gris.
- une ligne contient la largeur de l 'image, le caractère espace et la hauteur de l'image,
- une ligne contient la valeur maximale utilisée pour coder les niveaux de gris (255 pour notre cas)
- la suite du fichier est constituée d'une succession de valeurs représentant les pixels de l'image. Chaque pixel est codé par une valeur en caractères ASCII, l'image est codée ligne par ligne en partant du haut, chaque ligne est codée de gauche à droite,
- toute ligne commençant par le caractère # correspond à un commentaire.



```
P2
256 256
255
76 76 73 74 77 72 74 72 71 76 ...


.
.
.
.
.
```

Fig. 1 Image au format PNM P2

Génération de l'erreur : Afin de pouvoir visualiser l'effet du canal bruité, on ne perturbera pas l'entête du fichier PGM (« nombre magique », dimensions et niveau maximum de gris), seuls les pixels seront modifiés. Ecrire le programme *generreur* qui à partir d'un fichier au format PGM génère un nouveau fichier où, pour chaque pixel de l'image initiale, un bit a été modifié.

Encodage de l'image : Avant de transmettre l'image via le canal perturbé, il va falloir l'encoder. Pour cela on va utiliser le code binaire (7,4,3) de Hamming. Ce dernier permet d'encoder 4 bits en 7 bits et assure la correction d'une erreur. Chaque pixel de l'image PGM est représenté par un entier entre 0 et 255, ce qui correspond à 1 octet, soit 8 bits. On encodera donc l'image selon l'algorithme décrit dans la figure 2. Ecrire le programme *encodepgm* qui à partir d'une image au format PGM produit un fichier où tous les pixels ont été encodés en utilisant le code de Hamming binaire.

```
Pour chaque pixel p de l'image
        p1 ← 4 bits de poids fort de p
        p2 ← 4 bits de poids faible de p
        encoder p1
        encoder p2
```

Fig. 2 Algorithme d'encodage d'une image PNM

Décodage de l'image : Ecrire un programme *decodepgm* qui, à partir d'un fichier PGM encodé par l'algorithme de la figure 2, restitue le fichier PGM original. Il ne s'agit pas ici de corriger les éventuelles erreurs mais simplement de restituer le message d'information de 4 bits représenté par chaque mot de code.

Correction des erreurs : L'objectif est de corriger une image PGM encodée (via *encodepgm*) ayant subi des perturbations lors de son envoi. Ces perturbations seront simulées à l'aide de *generreur*.
Attention : *generreur* produit 1 bit d'erreur tous les 8 bits puisque les pixels de l'image d'origine correspondent aux entiers de 0 à 255. Les entiers présents dans le fichier PGM encodé sont dans l'intervalle [0,127] puisque le code de Hamming est de longueur 7. Ceci dit, lors de la lecture par *generreur,* du fichier PGM encodé, chaque entier lu a été stocké dans un octet. Ainsi deux cas de figure sont à considérer :

- l'erreur s'est produite sur le bit de poids fort. Il n'y a rien à faire puisque ce bit n'est pas utilisé lors du décodage (ce bit ne doit donc pas intervenir lors du calcul du syndrome),
- l'erreur s'est produite sur les 7 autres bits (donc sur le mot de code). Il faut corriger l'erreur avant de décoder.

Ecrire le programme *corrigepgm* qui, à partir d'un fichier PGM encodé contenant des erreurs, restitue le fichier PGM encodé initial.