

## Série de Tavaux Pratiques N°1

### Mise en œuvre d'un Codec RLE

#### Projet.

Il s'agit de mettre en œuvre un codec (codeur/décodeur) basée sur la technique RLE. Un pixel a deux valeurs possibles (0 pour le noir et 255 pour le blanc) et donc codé sur un octet : 00H ou FFH.

Une image de **n** pixels sera donc représentée par une chaîne de **n** caractères. Il s'agit de compresser et de décompresser une telle image. Pour cela, le codage suivant est utilisé :

- Si le même pixel se répète 3 fois ou plus, on utilisera un mot de deux octets : le bit de poids fort est à 1, et les 15 autres bits servent à coder le nombre de répétitions du pixel. Ces deux octets sont ensuite suivis de la couleur répétée.
- Si on a une suite de pixels différents (ou qui ne se répètent pas plus de 2 fois), on fera précéder cette suite d'un mot de deux octets où le bit de poids fort est à 0, et les 15 autres bits servent à coder la taille de la suite.

A titre d'exemple, les codes obtenus par cette méthode pour les images (6x6) pixels de la figure 1 sont les suivants :

- a) (0024H) 00H FFH 00H FFH ....00H FFH, soit  $2+36=38$  octets
- b) (8003H) 00H (8006H) FFH (8006H) 00H (8003H) FFH (8003H) 00H (8006H) FFH (8006H) 00H (8003H) FFH, soit  $3x8=24$  octets
- c) (800CH) 00H (800CH) FFH (800CH) 00H, soit 9 octets

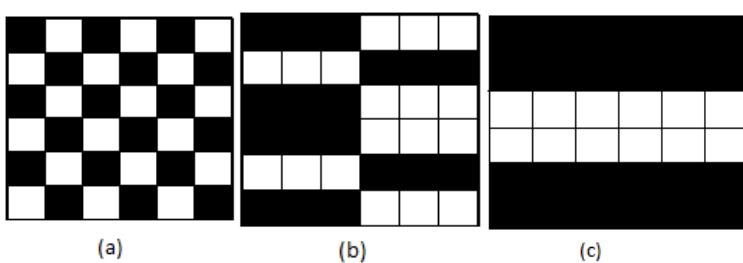


Figure 1. Exemple d'images

Deux programmes de manipulation d'images en python sont joints. Ils permettent de vous initier à la programmation en python.

### Tâches à réaliser :

- 1- Ecrire le programme en **Python** qui permet de lire une image (avec deux couleurs : Noir et Blanc) et produit un fichier de codes représentant l'image compressée. Calculer les taux de compression obtenu.
- 2- Ecrire un second programme en **Python** qui permet de lire un fichier compressé et de retrouver l'image d'origine et de la visualiser.