



Énigme 04 (première partie)

Compression des données

Dans le monde numérique qui nous entoure, toutes les données (images, films, sons, textes, etc.) sont codées informatiquement sous la forme d'une succession de 0 et de 1. Ce codage n'est pas limité au stockage sur des ordinateurs, il est aussi central pour les communications sur internet (courriels, « streaming » vidéo, etc.), pour les télévisions, les smartphones, etc.

Pour stocker, manipuler, envoyer toutes ces données numériques, il est nécessaire de les **compresser**, c'est-à-dire de les écrire de la manière la plus compacte possible.



Pourquoi compresser des données ?

Pour **optimiser la capacité de stockage** !

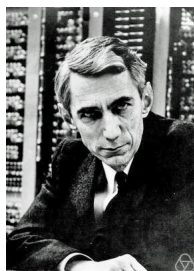
C'est pour cela que l'on peut enregistrer autant de photos sur nos smartphones.

Pour **optimiser la vitesse de transfert des fichiers** !

C'est pour cela que l'on peut regarder des vidéos en streaming sans qu'elles soient saccadées.

Claude Shannon (1916-2001)

Ingénieur et mathématicien américain qui a élaboré la théorie de la numérisation et de la compression de données analogiques.



Son impact sur notre société est absolument colossal car plusieurs milliards d'être humains utilisent chaque jour des données numériques compressées à travers nos outils numériques.

Compression RLE (Run-Length Encoding)

C'est le nom de l'algorithme de compression décrit ci-contre. Il est utilisé par les fax et aussi, combiné à d'autres techniques, par la compression des images au format JPEG, celui utilisé par nos smartphones pour enregistrer nos photos. Selon les données à compresser, il peut néanmoins se révéler peu efficace.

Compresser une image

Une image est mémorisée par des nombres. Pour simplifier, on ne considérera que des images en noir et blanc. Ainsi, chaque pixel sera soit noir, soit blanc.

La lettre « a » ci-contre a été agrandie pour bien voir les pixels. Si on représente les pixels noirs par un 0 et les blancs par un 1, il faut 30 caractères pour la mémoriser :

■	■	■	■	■	■
■	■	■	■	■	■
■	■	■	■	■	■
■	■	■	■	■	■
■	■	■	■	■	■
■	■	■	■	■	■

1, 0, 0, 0, 1
1, 1, 1, 1, 0
1, 0, 0, 0, 0
0, 1, 1, 1, 0
0, 1, 1, 1, 0
1, 0, 0, 0, 0

30
caractères

Compression possible

■	■	■	■	■	■
■	■	■	■	■	■
■	■	■	■	■	■
■	■	■	■	■	■
■	■	■	■	■	■
■	■	■	■	■	■

1, 3, 1
4, 1
1, 4
0, 1, 3, 1
0, 1, 3, 1
1, 4

17
caractères

La première ligne contient un pixel blanc, puis trois noirs puis un blanc. Ainsi, elle peut être représentée par 1, 3, 1.

Le premier nombre représente toujours le nombre de pixels blancs. Si le premier pixel est noir, la ligne commencera par un 0 (car il y a zéro pixel blanc pour commencer).

Avec cet algorithme, il n'y a besoin que de 17 caractères pour mémoriser l'image. On obtient un taux de compression égal à 43 % :

$$\text{taux de compression} = \frac{\text{taille} - \text{taille compressée}}{\text{taille}} = \frac{30 - 17}{30} \approx 0,43$$

Code à trouver

On donne ci-contre la compression d'une image avec la méthode décrite Au-dessus.

Le code de la première partie de cette énigme est la lettre correspondant à cette image, suivie du taux de compression obtenu, exprimé en pourcentage arrondi à l'unité.

Attention, on supprimera les espaces dans le code attendu. Exemple de réponse : a27.

1, 4, 1
0, 1, 4, 1
0, 1, 5
0, 1, 5
0, 1, 4, 1
1, 4, 1