

L'entier 10^8 a pour représentation binaire 10111101011110000100000000. C'est bien difficile à lire ! De la même façon qu'on peut séparer les chiffres par groupes de 3 dans l'écriture décimale d'un nombre (par exemple, 100 000 000), on choisit de séparer les bits de la représentation binaire par groupes de 8 : 101 11110101 11100001 00000000. Une séquence de 8 bits est alors appelée un *octet* : c'est cette unité qu'on utilise souvent pour évoquer la capacité mémoire de disques dans le commerce (256 Go signifie par exemple 256 Giga octets, soit 256 milliards d'octets, ce qui représente donc $256 \times 8 = 2048$ milliards de bits).

Exercice 1 On cherche à graver sur un bluray un flux vidéo non compressé. On suppose que le flux vidéo est de 1920×1080 pixels, que chaque pixel est codé sur 3 octets (codage rgb), qu'il y a 30 images par secondes et qu'un bluray peut stocker 50 Go de données.

1. Quelle est la durée maximale de vidéo que l'on peut stocker sur le bluray ? Qu'en conclure ?
2. On souhaite transmettre le flux vidéo via une connexion wifi d'un débit de 100 Méga bits par seconde. Quel est le nombre d'images que l'on peut transmettre par seconde via cette connexion ?

Exercice 2 Environ 25 000 étudiants sont inscrits à l'UFR Sciences de l'université d'Aix-Marseille. Un numéro est attribué à chaque étudiant. Bien évidemment, deux étudiants différents ne doivent pas avoir le même numéro.

1. Combien de bits sont nécessaires pour coder un de ces numéros ?
2. Les logiciels utilisés ont l'octet comme unité de mémoire. Combien d'octets sont nécessaires pour coder un numéro d'étudiant ?
3. L'administration de l'université souhaite que les programmes mis au point pour gérer l'inscription et la scolarité des étudiants puissent servir 10 ans. Si l'on pense que la fréquentation de l'université peut augmenter au plus de 20% chaque année, combien d'octets faut-il réserver pour coder les numéros des étudiants ?

Exercice 3 Dans cet exercice, nous allons considérer les deux codages de la figure 1 en page 2 pour les 27 symboles d'un texte (les 26 lettres de l'alphabet plus l'espace) qui associe à chaque symbole un mot binaire.

Par exemple le mot « patate » se code 010001010011010100110110 avec le codage variable et 011110000010011000001001100100 avec le codage fixe. Il est à noter que les espaces entre les codes des différentes lettres sont présents uniquement pour la lisibilité et ne font pas partie du code.

1. Coder votre nom avec les deux codages. Quel est le codage qui utilise le plus de bits ?
2. Décoder le message 0111011001001000001110110 avec les deux codages.
3. Décoder le message 110100010001101010000011001100000001101011010 avec les deux codages.
4. Décoder le message 100010010111100000001010010110100110100000001001 avec le codage variable.
5. Si on change le troisième bit du message de la question précédente que se passe-t-il ? Que devient le message si on change le deuxième bit ? le douzième bit ?
6. D'après vous, quels sont les avantages et les inconvénients du codage variable par rapport au codage fixe ?

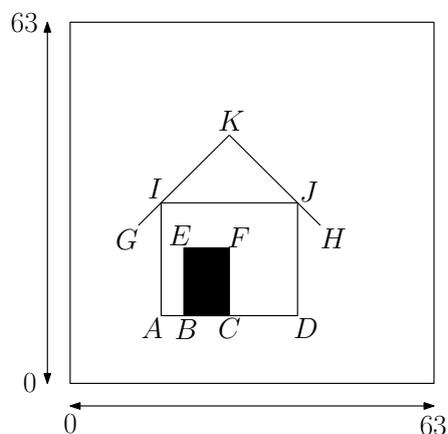
Exercice 4 (codage d'un dessin) On souhaite effectuer des dessins sur une grille carrée comprenant 64×64 cases (ou pixels). Chaque point est repéré par un couple d'entiers (x, y) où x et y sont compris entre 0 et 63. On supposera que x et y sont écrits en binaire.

1. Combien faut-il de bits pour coder un point ?

| lettre | codage variable | lettre | codage fixe |
|---------------|-----------------|---------------|-------------|
| <i>a</i> | 1010 | <i>a</i> | 00000 |
| <i>b</i> | 0010011 | <i>b</i> | 00001 |
| <i>c</i> | 01001 | <i>c</i> | 00010 |
| <i>d</i> | 01110 | <i>d</i> | 00011 |
| <i>e</i> | 110 | <i>e</i> | 00100 |
| <i>f</i> | 0111100 | <i>f</i> | 00101 |
| <i>g</i> | 0111110 | <i>g</i> | 00110 |
| <i>h</i> | 0010010 | <i>h</i> | 00111 |
| <i>i</i> | 1000 | <i>i</i> | 01000 |
| <i>j</i> | 011111110 | <i>j</i> | 01001 |
| <i>k</i> | 01111111001 | <i>k</i> | 01010 |
| <i>l</i> | 0001 | <i>l</i> | 01011 |
| <i>m</i> | 00101 | <i>m</i> | 01100 |
| <i>n</i> | 1001 | <i>n</i> | 01101 |
| <i>o</i> | 0000 | <i>o</i> | 01110 |
| <i>p</i> | 01000 | <i>p</i> | 01111 |
| <i>q</i> | 0111101 | <i>q</i> | 10000 |
| <i>r</i> | 0101 | <i>r</i> | 10001 |
| <i>s</i> | 1011 | <i>s</i> | 10010 |
| <i>t</i> | 0110 | <i>t</i> | 10011 |
| <i>u</i> | 0011 | <i>u</i> | 10100 |
| <i>v</i> | 001000 | <i>v</i> | 10101 |
| <i>w</i> | 01111111000 | <i>w</i> | 10110 |
| <i>x</i> | 01111110 | <i>x</i> | 10111 |
| <i>y</i> | 011111111 | <i>y</i> | 11000 |
| <i>z</i> | 0111111101 | <i>z</i> | 11001 |
| <i>espace</i> | 111 | <i>espace</i> | 11010 |

FIGURE 1 – Codage variable et codage fixe.

On suppose que les dessins seront composés de trois types d'éléments : des segments, des rectangles dont les côtés sont parallèles aux axes et qui pourront être pleins ou vides. Ainsi, le dessin de la figure ci-dessous est composé d'un rectangle vide $ADJI$, d'un rectangle plein $BCFE$ et de deux segments GK et KH .



— Un segment AB dont les extrémités ont pour coordonnées (x_A, y_A) et (x_B, y_B) sera repré-

senté par 4 octets $01x_A 01y_A 01x_B 01y_B$.

- Un rectangle vide $ABCD$ dont deux sommets opposés sont A et C sera représenté par 4 octets $10x_A 10y_A 10x_C 10y_C$.
- Un rectangle plein $ABCD$ dont deux sommets opposés sont A et C sera représenté par les 4 octets $11x_A 11y_A 11x_C 11y_C$.
- Un dessin est représenté par la suite des représentations des éléments qui le compose.

On remarque donc qu'on peut savoir si un octet code un élément d'un segment s'il commence par 01, un élément d'un rectangle vide s'il commence par 10 et un élément d'un rectangle plein s'il commence par 11.

2. Combien faut-il de bits pour représenter le dessin de la figure ? Combien cela fait-t-il en octets ?
3. Indiquez quel dessin est représenté par le codage suivant : 10001111 10001111 10101111 10101111 01001111 01001111 01101111 01101111 01001111 01101111 01001111.

On suppose que les $64 \times 64 = 4096$ pixels de la grille sont numérotés selon un ordre conventionnel (par exemple de gauche à droite et de haut en bas). Dans une représentation *bitmap*, un dessin (en noir et blanc) est représenté par une suite de bits $b_1 \dots b_n$ dont le i -ième bit b_i est égal à 0 si le i -ième pixel du dessin est blanc et 1 s'il est noir.

4. Combien faut-il d'octets pour représenter le dessin de la figure dans une représentation *bitmap* ?

On souhaite enrichir les codages possibles en représentant directement des lignes brisées $A_1A_2 \dots A_k$ (c'est-à-dire des réunions de segments $A_1A_2, A_2A_3, \dots, A_{k-1}A_k$).

5. Indiquez comment on pourrait étendre le codage ci-dessus pour représenter de telles lignes brisées ? (*Indication : les extrémités de la ligne pourront être repérées par 01 et les points intérieurs par 00. Combien faudrait-il alors de bits pour représenter le dessin de la figure ?*)

Exercice 5 Le cerveau humain a environ 100 milliards de neurones qui ont chacun en moyenne 10 000 synapses transmettant environ 100 impulsions binaires par seconde. Calculer la quantité d'information maximale transmise par seconde dans un cerveau, en octets.