

## Stockage des données

**Exercice 1.** Quel type de croissance est décrit par la seconde loi de Moore ?

**Exercice 2.** Avant d'être mesurées par des puissances de 10 d'octets (ko, Mo, Go, etc), les données informatiques étaient mesurées en puissances de 2 d'octets. Pendant longtemps, un kilooctet a désigné  $2^{10}$  octets, un mégaoctet a désigné  $2^{20}$  octets, un gigaoctet a désigné  $2^{30}$  octets et ainsi de suite. Cependant, cette tradition propre au domaine de l'informatique entraine en conflit avec les normes internationales (selon lesquelles 1 kilo correspond à  $10^3$ , un méga à  $10^6$ , etc). Ainsi, en 1988, la Commission électrotechnique internationale a normalisé les unités en introduisant des préfixes spécifiques pour les puissances de 2. Ainsi, sont apparus les termes *kilo binaire*, *méga binaire*, *giga binaire*, *téra binaire* et *péta binaire* abrégé en kibi, mébi, gibi, tébi et pébi. On a alors le tableau suivant :

| unité     | kibioctet (Kio) | mébioctet (Mio) | gibioctet (Gio) | tébioctet (Tio) | pébioctet (Pio) |
|-----------|-----------------|-----------------|-----------------|-----------------|-----------------|
| en octets | $2^{10}$        | $2^{20}$        | $2^{30}$        | $2^{40}$        | $2^{50}$        |

1. Les premières disquettes 3 1/2 construites par Sony avaient une capacité de 400 Kio. Déterminer la capacité en ko d'une telle disquette.
2. Un disque dur S-ATA Hitachi de fin 2005 avait une capacité de stockage de 76,688 Gio. Convertir cette capacité en Go.

**Exercice 3.**

1. Quelle la taille en octet d'un fichier texte codé en ASCII et contenant le texte suivant ?

Je dois déterminer la taille de ce texte.  
 Pour cela, je ne dois pas oublier les espaces et la ponctuation,  
 ni les retours à la ligne.

2. Un fichier texte codé en ASCII compte 12 lignes. Chaque ligne compte 30 caractères (espaces et ponctuation compris). Quelle est la taille en octet de ce fichier ?
3. Un fichier texte codé en ASCII a une taille de 236 ko. Déterminer le nombre maximum de caractères qu'il peut contenir.

**Exercice 4.** Dans un dossier, on trouve les fichiers suivants :

fichier1.jpg   fichier2.txt   fichier3.mov   fichier4.exe   fichier5.png  
 fichier6.mp4   fichier7.doc   fichier8.avi   fichier9.wav   fichier10.mp3

Regrouper ces fichiers en 5 catégories : fichiers texte, fichiers image, fichier son, fichiers vidéo et fichiers exécutables.

**Exercice 5.** On dispose 3 fichiers `fichier1`, `fichier2` et `fichier3`. On sait que la taille de `fichier1` est 840 Mo, la taille de `fichier2` est 53 Mo et la taille de `fichier3` est 14 ko. On sait également que les extensions de ces fichiers sont `.txt`, `.wav` et `.avi`.

En se référant aux ordres de grandeurs standards, déterminer l'extension de chaque fichier.

**Exercice 6.** (Source : <http://images.math.cnrs.fr/Le-traitement-numerique-des-images.html>)

Lorsqu'on numérise une image en niveau de gris, on stocke cette image sous la forme d'un tableau constitué de petits carrés appelés pixels. À chaque pixel, on va associer un nombre entre 0 et 255 correspondant à un certain niveau de gris comme sur l'image suivante :



Image A

Chaque nombre est ensuite codé sur un octet comme dans le cas du code ASCII.

1. L'image A a une résolution de 240 par 240, c'est-à-dire qu'elle correspond à un tableau ayant 240 lignes et 240 colonnes. Combien de pixels constituent cette image ? Déterminer sa taille en ko.
2. Pour gagner de la place, on peut soit diminuer le nombre de pixels soit diminuer le nombre de niveaux de gris.



Image B



Image C

- a. Déterminer la taille de l'image B sachant qu'elle a été obtenue à partir l'image A en ne conservant qu'une ligne sur deux et qu'une colonne sur deux.
- b. Déterminer la taille de l'image C sachant qu'elle a été obtenue à partir l'image A en ne conservant que 16 niveaux de gris au lieu de 256.

**Exercice 7.**

1. Écrire une fonction Python `en_binaire(N)` qui prend en argument un entier `N` compris en 0 et 255 et qui renvoie l'octet (sous forme d'une chaîne de caractères) qui code cet entier en écriture binaire.
2. Écrire une fonction Python `en_decimal(O)` qui prend en argument un octet `O` (sous forme d'une chaîne de caractères) et qui renvoie l'entier codé par `O` en binaire.
3. Déterminer le code ASCII binaire correspondant au mot `Binaire`.
4. À quel mot correspond le code ASCII binaire suivant :

01000100 01100101 01100011 01101001 01101101 01100001 01101100