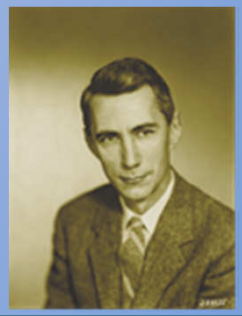
Introduction

Pour décrire une image, commence par comprendre comment ton œil la voit.

C'est encore avec des 0 et des 1 que l'on représente les images et les sons, mais en grand nombre.
Pour décrire une image, on peut utiliser une représentation **vectorielle** en décrivant des formes géométriques : un cercle, une droite, etc., ou une représentation **bitmap** en quadrillant l'image et en décrivant chaque case (**pixel**).
Noir et blanc, niveaux de gris ou couleurs sont décrits par différents codages.

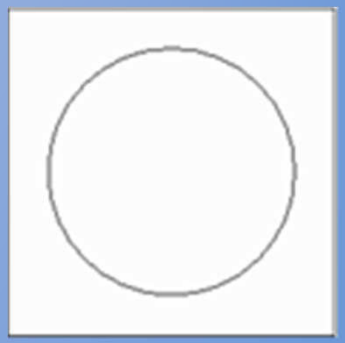
Les sons aussi sont échantillonnés en découpant le temps durant lequel le son est émis.
Les images et les sons sont de très longues suites de 0 et de 1, nous introduisons dans ce chapitre les unités pour mesurer la taille des fichiers.



Claude Shannon (1916 - 2001), a montré en 1949, en s'appuyant sur des travaux antérieurs de Harry Nyquist, que la fréquence d'échan- tillonnage d'un son, et plus générale- ment d'un signal, doit être au moins le double de la fréquence maximale contenue dans ce son, pour que le son puisse être restitué à partir de l'échantillon.
Il a également montré comment décrire les circuits électroniques par des fonctions booléennes et comment exprimer toutes les fonctions booléennes et arithmétiques à l'aide des fonctions booléennes

La représentation des images

Représentation vectorielle d'une image :



Pour décrire l'image ci-contre , une possibilité est de dire :
« Cette image est formée d'un cercle ».
On peut même être plus précis et indiquer les coordonnées du centre du cercle et son rayon. Et, à partir de cette description, n'importe qui pourrait reconstituer le dessin.

On peut donc représenter cette image par trois nombres : deux pour les coordonnées du centre et un pour le rayon. Une image formée de plu sieurs cercles serait de même décrite par trois nombres pour chacun d'eux. On peut représenter d'une manière similaire un dessin formé de cercles et de rectangles, en représentant chaque figure par une lettre,
« c » pour un cercle, « r » pour un rectangle, suivi d'une suite de nombres qui définissent les paramètres de la figure. On parle alors de **représentation symbolique**, ou parfois de **représentation vectorielle**, d'une image.

La représentation des images

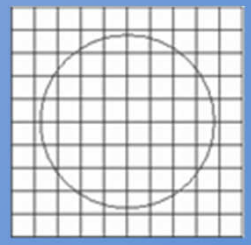
Représentation en pixels d'une image :

Néanmoins, cette méthode est peu pratique pour représenter l'image ci-contre.

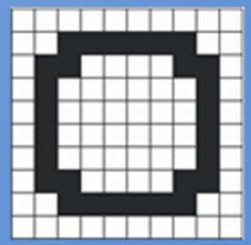
Une autre méthode, qui a l'avantage de pouvoir être utilisée sur n'importe quelle image, consiste à superposer un quadrillage à l'image.

Chacune des cases de ce quadrillage s'appelle un pixel (picture element).

On noircit ensuite les pixels qui contiennent une portion de trait.



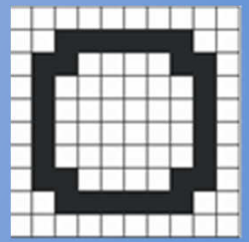
Puis, il suffit d'indiquer la couleur de chacun des pixels, en les lisant de gauche à droite et de haut en bas, comme un texte. Ce dessin se décrit donc par une suite de mots « blanc » ou « noir ».



Comme seuls les mots « noir » ou « blanc » sont utilisés, on peut être plus économe et remplacer chacun de ces mots par un bit, par exemple 1 pour « noir » et 0 pour « blanc ».

La représentation des images

Représentation en pixels d'une image :



Le dessin ci-contre, avec une grille de 10 × 10, se décrit alors par la suite de 100 bits suivante :

0000000000001111110001100001100100000010010000001001000000100100000010011000011000111111000000000000

Cette description est assez approximative, mais on peut la rendre plus pré cise en utilisant un quadrillage, non plus de 10 × 10 pixels, mais de 100 × 100 pixels.

À partir de quelques millions de pixels, notre œil n'est plus capable de faire la différence entre les deux images.

Cette manière de représenter une image sous la forme d'une suite de pixels, chacun exprimés sur un bit, s'appelle une **bitmap**.

C'est une méthode approximative, mais universelle : n'importe quelle image en noir et blanc peut se décrire ainsi.

La notion de format

Les textes, images, un son, etc. par exemple sur un disque, on le stocke dans un fichier. Pour le moment, nous avons juste besoin de savoir qu'un fichier est un mot, c'est-à-dire une suite de 0 et de 1, auquel on a attribué un nom.

Si on trouve un fichier qui contient la suite de bits

0000000000001111110001100001100100000010010000001001000000100100000

0100100000010011000011000011111100000000000000

il n'est pas a priori évident que cette suite exprime une image de 10×10 pixels.

Il pourrait aussi s'agir par exemple d'un texte en ASCII, ou de la représentation d'un nombre à virgule.

Pour cette raison, au lieu d'appeler le fichier simplement cercle, on l'appelle **cercle.pbm**. Cette extension pbm du nom indique que ce fichier est exprimé dans le format **PBM** (Portable **BitMap**). Ce format est l'un des plus simples pour exprimer des images.

La notion de format

Un fichier au format PBM est un fichier en ASCII qui se compose comme suit :

- les caractères P1, suivis d'un retour à la ligne ou d'un espace,
- la largeur de l'image, en base 10, suivie d'un retour à la ligne ou d'un espace,
- la hauteur de l'image, en base 10, suivie d'un retour à la ligne ou d'un espace,
- la liste des pixels, ligne par ligne, de haut en bas et de gauche à droite.

Toutes les lignes commençant par le caractère # sont des commentaires ignorés.

Ainsi, le code ci-contre est un fichier au format PBM.

```
P1
# Mon  premier  fichier
PBM  : cercle
10 10
0000000000
0011111100
0110000110
0100000010
0100000010
0100000010
0100000010
0110000110
0011111100
0000000000
```

La notion de format

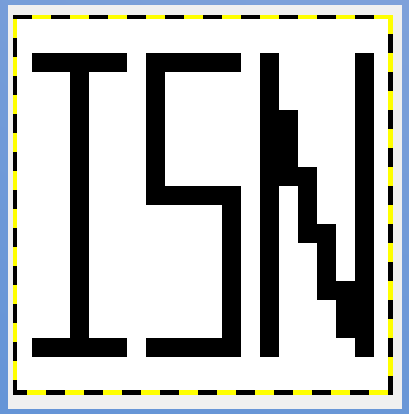
Exercice 1

Créer avec un éditeur de texte simple (comme Notepad) un fichier au format PBM de dimensions 20x20 qui affiche le logo ISN.

A quel problème d'affichage est-on confronté ?

Correction exercice 1

On est confronté à un problème de « crénelage » pour l'affichage de la lettre N.




```
P1
20 20
00000000000000000000
00000000000000000000
01111101111101000010
00010001000001000010
00010001000001000010
00010001000001100010
00010001000001100010
00010001000001100010
00010001000001110010
00010001111101010010
00010000000101010010
00010000000101011010
00010000000101001010
00010000000101001010
00010000000101001110
00010000000101000110
00010000000101000110
01111101111101000010
00000000000000000000
00000000000000000000
```

La notion de format

Voici une liste non exhaustive des principaux formats d'images :

- PBM (Portable **B**it**M**ap), PGM (Portable Gray**M**ap), PPM (Portable **P**ix**M**ap)
- GIF (Graphics Interchange Format) : format présent sur le WEB possibilité d'animer l'image (gif animé)
- PICT : format créé par Apple pour le Macintosh II
- BMP : (BitMaP) : format créé pour les PC
- JPEG (JPEG File Interchange Format) : créé avec l'arrivée de la photographie numérique **Jpeg** (***J**oint **P**hotographic **E**xperts **G**roup*)
- TIFF (***T**agged **I**mage **F**ile **F**ormat**) : format utilisé par les imprimeurs*
- PNG (Portable **N**etwork **G**raphics) : créé pour remplacer le GIF (utilisé par les graphistes notamment)
- EPS (**E**ncapusulated **P**ost**S**cript) : format vectoriel
- SWF (**S**hock**W**ave **F**lash) : format d'animation ou d'image vectorielles utilisé sur Internet par le logiciel Flash..
- SVG (Scalable Vector Graphics) : format de données conçu pour décrire des ensembles de graphiques vectoriels et basé sur XML

La notion de format

SAVOIR-FAIRE Identifier quelques formats d'images

Les différents formats qui permettent d'exprimer des images se distinguent les uns des autres par :

- le type d'image : noir et blanc, en niveaux de gris, en couleurs, etc. ;
- la manière d'exprimer ces images : sous forme vectorielle ou de bitmap ;
- le fait que ces images soient compressées ou non ;
- le fait que le format soit public ou secret : certains formats sont publics, d'autres sont gardés secrets par leurs concepteurs, de manière à contraindre les utilisateurs à utiliser leurs logiciels pour traiter ces images ;
- le fait que ces formats soient propriétaires ou libres : pour utiliser certains formats, il est nécessaire de payer des droits à leurs concepteurs, alors que pour d'autres non.

La notion de format

Exercice 2

Chercher sur le Web les caractéristiques des formats PBM, GIF, JFIF, PNG et SVG.

Correction exercice 2

- PBM (*Portable BitMap) :* format noir et blanc, bitmap, non compressé, public et libre.
- GIF (Graphics Interchange Format) : format couleur, bitmap, compressé, codé sur 8 bits limité à 256 couleurs, propriétaire initialement et libre depuis 2006.

  Le format GIF89a permet de stocker des images animées.
- JFIF (JPEG File Interchange Format) : format couleur, bitmap, compressé avec perte de qualité, public et libre.
- PNG (Portable Network Graphics) : format couleur, bitmap, compressé sans perte de qualité, public et libre.
- SVG (Scalable Vector Graphics) : format couleur, vectoriel, basé sur XML, public et libre.

Images en niveaux de gris

Certaines images, par exemple les photos en noir et blanc, utilisent, en plus du noir et du blanc, diverses **nuances de gris**.

On les appelle les images en **niveaux de gris**.

Un format, parmi d'autres, pour exprimer ces images est le format **PGM** (**P**ortable **G**rey**M**ap).

Pour exprimer une image dans le format PGM, on choisit une valeur maximale, par exemple 255, pour exprimer les niveaux de gris et on associe à chaque pixel un nombre compris entre 0 et 255, 0 indiquant que le pixel est noir et 255 qu'il est blanc.

Les valeurs de 1 à 254 expriment différentes teintes de gris, de la plus foncée à la plus claire.

Un fichier au format PGM, ressemble beaucoup à un fichier au format PBM, c'est un fichier en ASCII qui se compose comme suit :

- les caractères P2, suivis d'un retour à la ligne ou d'un espace,
- la largeur de l'image, suivie d'un retour à la ligne ou d'un espace,
- la hauteur de l'image, suivie d'un retour à la ligne ou d'un espace,
- la valeur maximale utilisée pour exprimer les niveaux de gris, suivie d'un retour à la ligne ou d'un espace,
- la liste des couleurs des pixels, ligne par ligne, de haut en bas et de gauche à droite, séparées par des retours à la ligne ou des espaces.

Images en couleurs

Pour comprendre comment représenter les images en couleurs, il faut d'abord s'intéresser à la manière dont notre œil perçoit ces dernières.

Notre œil contient des cellules, les cônes, qui sont sensibles à la couleur, c'est-à-dire à la longueur d'onde de la lumière qu'ils reçoivent.

Ces cônes sont de trois sortes, dont le maximum de sensibilité est respectivement dans le rouge (560 nm), le vert (530 nm) et le bleu (424 nm).

Quand notre œil reçoit une lumière monochrome émise par une ampoule jaune, les cônes sensibles au rouge et au vert réagissent beaucoup et ceux sensibles au bleu un tout petit peu, exactement comme s'il recevait un mélange de lumières émises par deux ampoules rouge et verte.

Images en couleurs

Ainsi, en mélangeant de la lumière produite par une ampoule rouge et une ampoule verte, on peut donner à l'œil la même sensation que s'il recevait une lumière jaune.

Plus généralement, quelle que soit la lumière qu'il reçoit, notre œil ne communique à notre cerveau qu'une information partielle : l'intensité de la réaction des cônes sensibles au rouge, au vert et au bleu.

Et deux lumières qui stimulent ces trois types de cônes de manière identique sont indiscernables pour l'œil.

Ainsi, sur l'écran d'un ordinateur, chaque pixel est composé non pas d'une, mais de trois sources de lumière, rouge, verte et bleue ; en faisant varier l'intensité de chacune de ces sources, on peut simuler n'importe quelle couleur.

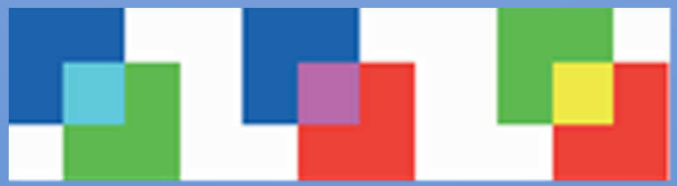
Images en couleurs

Par exemple, en mélangeant de la lumière verte et de la lumière bleue on obtient de la lumière cyan.

En mélangeant de la lumière rouge et de la lumière bleue on obtient de la lumière magenta.

Et en mélangeant de la lumière rouge et de la lumière verte on obtient de la lumière jaune.

« Cyan » est le nom savant d'un bleu turquoise et « magenta » celui d'un rouge tirant un peu sur le violet.



Un format, parmi d'autres, pour exprimer ces images est le format **PPM** (Portable PixMap).

Pour exprimer une image dans ce format, on choisit une valeur maximale, par exemple 255, pour exprimer l'intensité des couleurs et on associe à chaque pixel trois nombres, l'intensité en rouge, en vert et en bleu, chaque nombre étant compris entre 0 et 255.

Un fichier au format PPM, ressemble beaucoup à un fichier au format PBM ou PGM ; c'est un fichier ASCII qui se compose comme suit :

- les caractères P3, suivis d'un retour à la ligne ou d'un espace,
- la largeur de l'image, suivie d'un retour à la ligne ou d'un espace,
- la hauteur de l'image, suivie d'un retour à la ligne ou d'un espace,
- la valeur maximale utilisée pour exprimer l'intensité des couleurs,
- la liste des valeurs des couleurs, trois par pixel, dans l'ordre rouge, vert, bleu, ligne par ligne, de haut en bas et de gauche à droite, séparées par des retours à la ligne ou des espaces.

Par exemple, en mélangeant du rouge et du vert en quantités égales, on obtient du jaune.

En augmentant la quantité de rouge, par exemple rouge = 237, vert = 127, bleu = 16, on obtient du orange.

On peut alors écrire un fichier PPM qui représente un carré orange de 100 pixels sur 100 pixels, le triplet 237 127 16 devant être répété dix mille fois, puisque l'image est formée de dix mille pixels (voir ci-contre).

Cet exemple aide à comprendre pourquoi il y a des formats d'images plus complexes que le format PPM : il y a de nombreux moyens d'éviter de recopier dix mille fois le même triplet de nombres. C'est ce que l'on appelle **compresser** un fichier.

```
P3
# Mon premier fichier PPM : orange
100 100
255
237 127 16
237 127 16
237 127 16
237 127 16
...
```

SAVOIR-FAIRE Numériser une image sous forme d'un fichier

1. Identifier le type d'image à numériser (noir et blanc, en niveaux de gris ou en couleurs) et choisir un format approprié.
2. Identifier la valeur de chaque pixel.
3. Identifier les autres informations que contient le fichier : taille de l'image, commentaires, etc. et la manière dont ces informations sont exprimées dans ce format.

Exercice 3

Lequel des formats PBM, PGM et PPM est adapté pour représenter un carré noir de 10 pixels sur 10 pixels ?

Même question pour un carré rouge de même taille.

Comparer les tailles des fichiers obtenus.

Correction exercice 3

Pour un carré noir, le format PBM suffit, puisqu'il n'y a ni niveaux de gris ni couleurs, et le fichier est le suivant (voir ci-contre).

```
P1
# Un carré noir
10 10
1111111111
1111111111
1111111111
1111111111
1111111111
1111111111
1111111111
1111111111
1111111111
1111111111
```

Pour un carré rouge, il faut obligatoirement recourir au format PPM, seul capable de représenter de la couleur.

Le rouge se représente par les trois nombres 255 0 0 : intensité maximale pour le rouge et nulle pour le vert et le bleu. On obtient le fichier ci-contre.

```
P3
# Un carré rouge
10 10
255
255 0 0
255 0 0
255 0 0
…
```

Sans ligne de commentaire, le fichier pour le carré noir fait 131 octets et celui pour le carré orange fait 1216 octets soit environ 10 fois plus gros.

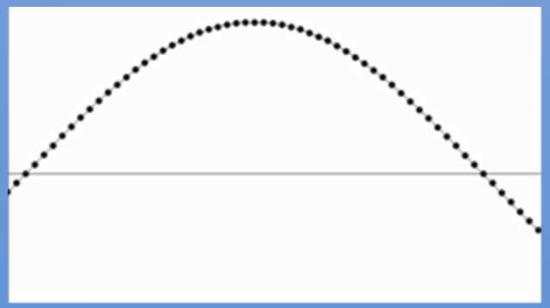
La représentation des sons

Un son est une variation de la pression de l'air au cours du temps.

Une manière simple de représenter un son consiste à l'**échantillonner**, c'est-à-dire mesurer la pression à intervalles réguliers et le représenter comme la suite des mesures obtenues.

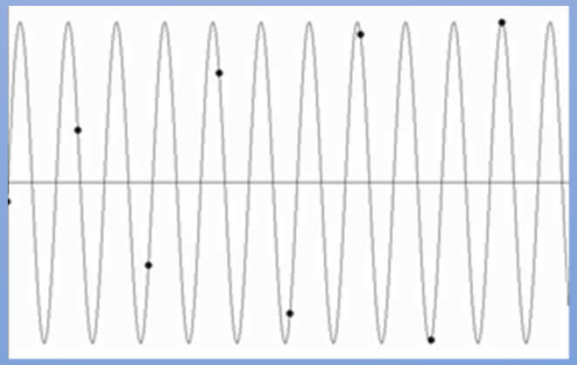
L'échantillonnage d'un son est une méthode assez similaire au découpage d'une image en pixels, sauf que le découpage s'effectue, non dans l'espace, mais dans le temps.

Par exemple, si on échantillonne à 44 000 Hz, c'est-à-dire en faisant 44 000 mesures par seconde, une sinusoïde de fréquence 440 Hz, on fait cent mesures par période et on obtient l'échantillon suivant :



La représentation des sons

En revanche, si l'on échantillonne à 300 Hz cette même sinusoïde, on fait une mesure toutes les périodes et demie environ, et on obtient l'échantillon suivant :



On comprend qu'il est possible de reconstituer la sinusoïde dans le premier cas, mais pas dans le second.

De manière plus générale, on montre que quand on échantillonne un son, il faut, pour que la reconstitution soit possible, une fréquence d'échantillonnage au moins double de la fréquence la plus élevée contenue dans ce son. (Théorème d'échantillonnage de Nyquist-Shannon)

La représentation des sons

En général, on échantillonne les sons à 44 000 Hz, car le son sinusoïdal le plus aigu que notre oreille peut entendre est de 22 000 Hz environ, ce qui implique que notre oreille ne peut pas distinguer deux sons qui donnent le même échantillon à 44 000 Hz.

Cette fréquence est relativement élevée ; c'est pourquoi il faut plusieurs millions de bits pour représenter une minute de son et les fichiers audio sont souvent compressés.

Cette méthode de représentation d'un son par échantillonnage est utilisée dans de nombreux formats.

Les plus simples sont RAW, WAV et BWF, mais il existe de nombreux autres formats plus sophistiqués : MP3, WMA, AAC, etc.

Comme pour les images, ces formats se distinguent les uns des autres par la manière dont le son est représenté, par le fait qu'il soit compressé ou non, que le format soit public ou secret et propriétaire ou libre.

La taille d'un texte, d'une image ou d'un son

La taille d'une suite de 0 et de 1, que cette suite représente un texte, une image ou un son, s'exprime soit en bits, soit en octets, un octet étant égal à 8 bits.

Par exemple, la suite 0111 0001 0010 0111 a une taille de 16 bits, ou encore de 2 octets.

Comme les textes, les images et les sons sont souvent de longues suites ; on peut exprimer leurs tailles en kilooctets, mégaoctets, gigaoctets ou téraoctets.

Comme en physique, un kilo (k) est un millier (10^3), un méga (M) un million (10^6), un giga (G) un milliard (10^9) et un téra (T) mille milliards (10^{12}).

Ainsi, une image d'un mégaoctet est formée de huit millions de bits.

On utilise cependant souvent des préfixes similaires qui expriment des nombres ronds, non en décimal, mais en binaire :

- un kilo (binaire) est 1 024 (2^{10}),
- un méga (binaire) 1 048 576 (2^{20}),
- un giga (binaire) 1 073 741 824 (2^{30}),
- un téra (binaire) 1 099 511 627 776 (2^{40}).

La taille d'un texte, d'une image ou d'un son

SAVOIR-FAIRE Comprendre les tailles des données et les ordres de grandeurs

Trois quantités peuvent entrer en jeu dans la taille d'un fichier de données :

1. le nombre de bits utilisé pour représenter une « unité » de donnée : pixel, caractère alphanumérique, échantillon sonore, etc. ;
2. éventuellement, selon le format, la fréquence d'échantillonnage : nombre d'échantillons par seconde, de pixels par centimètre, etc. ;
3. la taille de l'objet dans une unité concrète : durée du son en secondes, surface de l'image en centimètres carrés, etc.

Exercice 4

On enregistre un son pendant 10 min avec 10 000 échantillons par seconde et 16 bits pour chaque échantillon, sans compresser les données.

Quelle est la taille du fichier ?

Correction exercice 4

La taille du fichier est $10 \times 60 \times 10\,000 \times 16$ bits = 96 000 000 bits : 93 Mb (mégabits décimaux)

Ou 91,55 mégabits (binaires) environ. ($96\,000\,000 / 2^{20}$)

La taille d'un texte, d'une image ou d'un son

Exercice 5

On enregistre une image 10 cm × 10 cm, avec 100 pixels par centimètre, chaque pixel étant représenté par trois nombres entiers, chacun codé sur un octet.

Quelle est la taille du fichier ?

Correction exercice 5

La taille du fichier est : 10×10×100×3 octets = 30 000 octets = 30 Ko

La taille d'un texte, d'une image ou d'un son

SAVOIR-FAIRE **Choisir un format approprié par rapport à un usage ou un besoin, à une qualité, à des limites**

Identifier les points suivants.

1. Les données doivent-elles être stockées avec précision ou un certain taux de perte est-il acceptable ?
2. Quelle taille est acceptable pour le fichier ainsi créé ?
3. Quels logiciels devront pouvoir accéder au fichier ?

La taille d'un texte, d'une image ou d'un son

Exercice 6

1) Ouvrir un logiciel de traitement d'images tel que Gimp et y charger une photo. Observer dans le logiciel sa taille en mémoire en choisissant la fonction *Propriété de l'image* dans le menu *Image*.

Cette taille en mémoire est le produit du nombre de pixels et de la taille des pixels.

En déduire la taille, en octets, de chaque pixel. Pour une photo en couleurs, le résultat devrait être 3 octets. Toutefois, si le format de l'image n'est pas PPM, le résultat peut être différent.

2) Enregistrer cette image sous différents formats et à différents niveaux de compression.

Comparer visuellement les résultats obtenus. Observer si le taux de compression proposé correspond bien au rapport des tailles des fichiers.

La taille d'un texte, d'une image ou d'un son

Correction exercice 6

1) Image utilisée :

avec 4 formats différents

Format	jpg	ppm brut	ppm ASCII	gif
Taille sur disque	416 Ko	7 706 Ko	7706 Ko	589 Ko
Taille en mémoire	7 Mo	5,5 Mo	7 Mo	7 Mo
Dimensions en pixels	1024 x 768	1024 x 768	1024 x 768	1024 x 768
Nombre de pixels	786432	786432	786432	786432
Taille d'un pixel en octets = Rapport taille en mémoire/nombre de pixels	≈ 9	≈ 7	≈ 9	≈ 9

La taille d'un texte, d'une image ou d'un son

Correction exercice 6

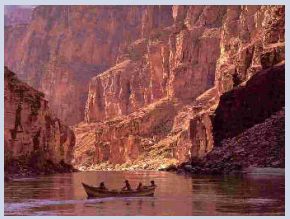
2) Utilisation d'une compression JPEG avec différents taux.

Image	Qualité	Taille disque	Rapport Taille/taille initiale
	100%	416 Ko	1
	95%	403 Ko	0,97
	50 %	109 Ko	0,26

La taille d'un texte, d'une image ou d'un son

Correction exercice 6

2) Utilisation d'une compression JPEG avec différents taux.

Image	Qualité	Taille disque	Rapport Taille/taille initiale
	10%	29 Ko	0,07
	5%	15 Ko	0,036