

Les fichiers informatiques

« numériques »

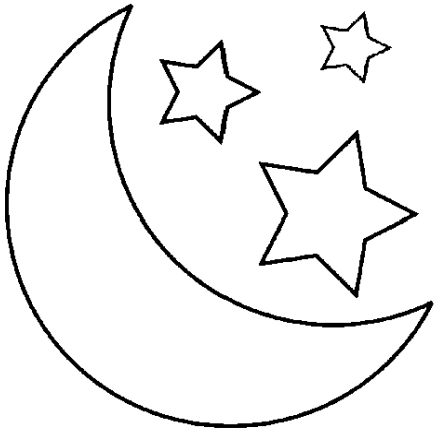
Requis 1

« Le langage de l'ordinateur ne comprend que 2 lettres (0 et 1) »

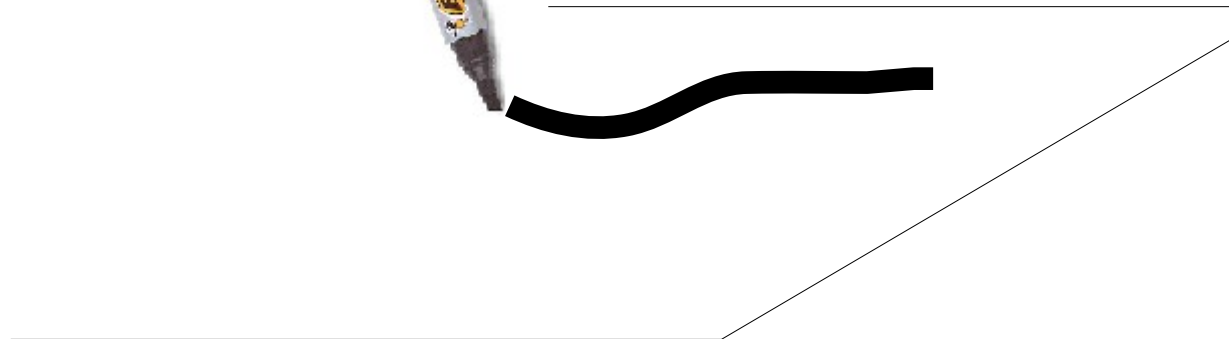
Cf : unité centrale

Pour exprimer

Des images



Activité préparatoire

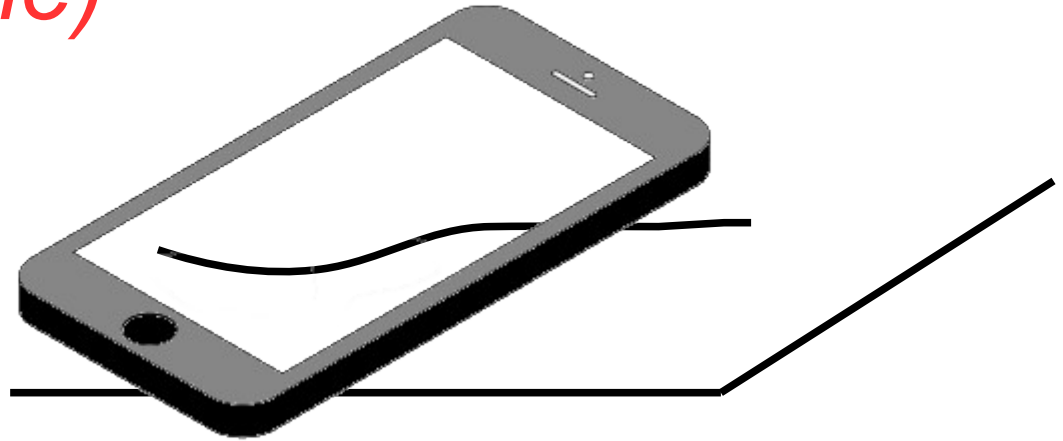


*Si je regarde avec une
très bonne loupe*



On voit un Trait continu

Prise d'une photographie (avec un smartphone)



*L'image est constituée de
petit carrés*

Une image informatique est décomposée en « points » juxtaposés que l'on appelle « pixel ».

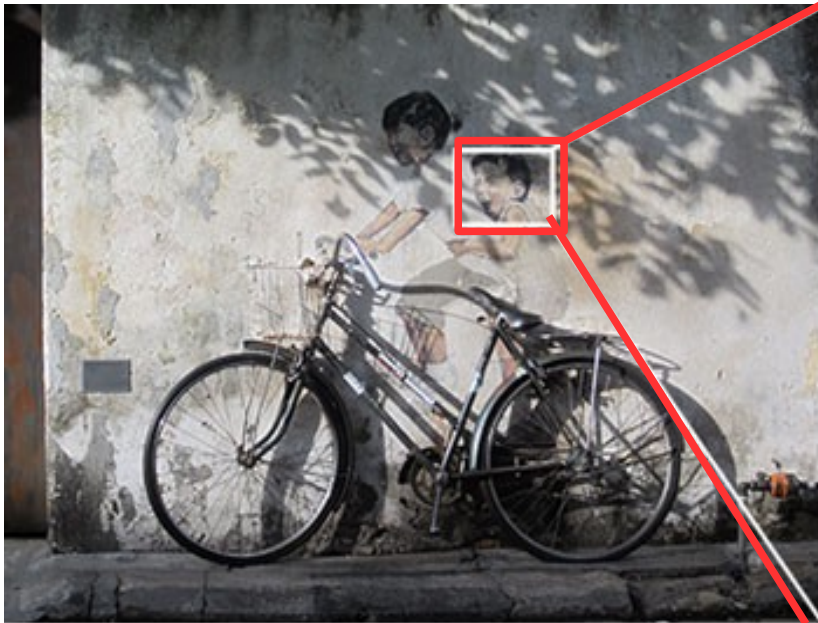


Image peinte (continue)

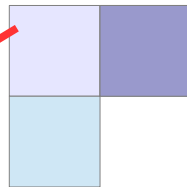
« Réalité »



« fichier Informatique »

Ensemble de pixel

3 pixels



Information analogique

numérisation



Information numérique

Pixel : « picture elements »

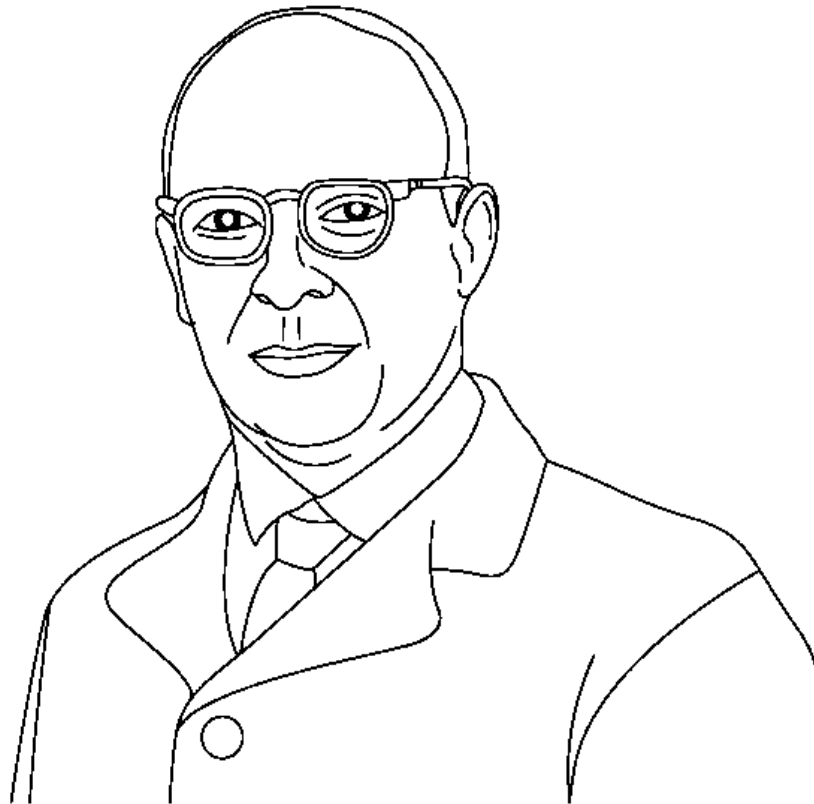
Cette « discrétisation » apparaît pour toutes grandeurs réelles que l'on veut enregistrer dans un fichier informatique.

Musique, évolution de température etc...

Comment coder (enregistrer) une image ?

En utilisant que 2 signes

0 et 1



Exemple : image en noir et blanc

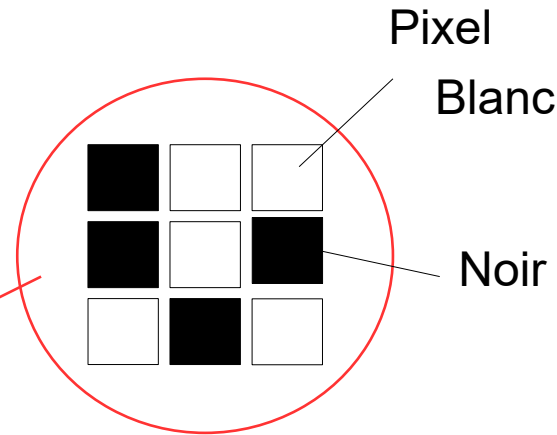
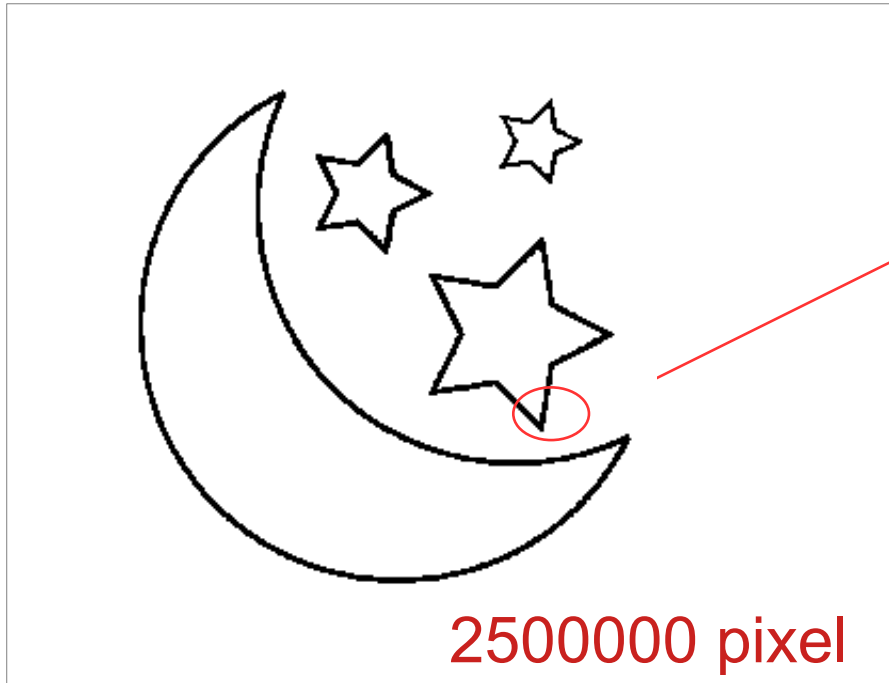
John
Bordeen



Fichier binaire de l'image ?

Indice : Activité découverte 2

Bit Map Picture monochrome



Taille mesurée 32062 octets
256496 bit

Chaque pixel semble être codé avec 1 bit

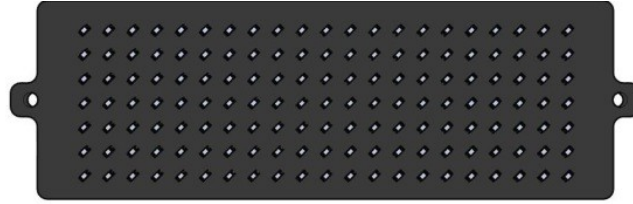
Il y a un petit écart : 6496 bit ou 812 octet

Indice :

Activité découverte robot weeebot



(Matrice a DEL)



Ecran monochrome » : 21 x 7 pixel

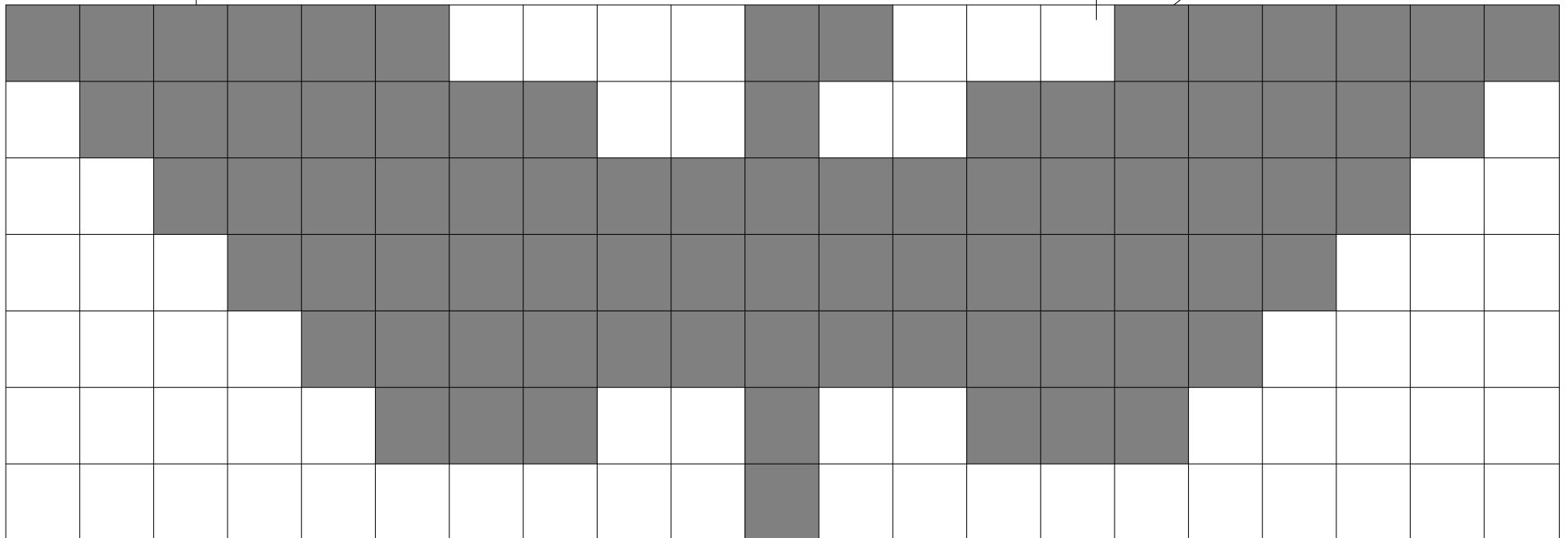
147 Diodes Électroluminescentes (DEL)

Peut afficher des images de 147 pixel monochrome

exemple **pixel**

blanc = éteint

noir =allumé



Fichier binaire de l'image ?

1	1	1	1	1	1	0	0	0	0	1	1	0	0	0	1	1	1	1	1	1
0	1	1	1	1	1	1	1	0	0	1	0	0	1	1	1	1	1	1	1	0
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	0	1	1	1	0	0	1	0	0	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

0 = éteint

1 = allumée

147 bit pour coder l'image

```
1111110000110001111110111111100100111111100011111111111
11111100000111111111111111110000000111111111111100000000
0111001001110000000000000000000000100000000000
```

→ codage binaire d'une image

La plus simple image format BMP en noir et blanc

Une image en **noir et blanc**

Chaque pixel est codé sur **1 bit**

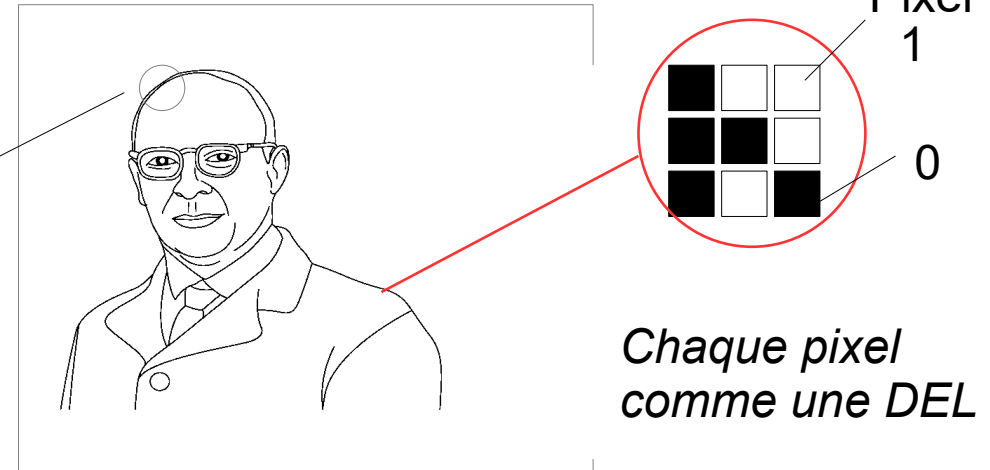
1 = allumé (blanc)

0 = éteint (noir)

Ainsi 480000 bits pour l'image 800x600

*...111111111111111110111111
11111.....11111111111111111011
111111111*

Bit Map Picture



*Chaque pixel
comme une DEL*

480 000 pixels

Soit 60 000 Octets

En fait taille affichée : 60062 octet : .

Option propriétés

Taile :

58,6 Ko (60 062 octets)

62 octets d'écart entre le codage de l'image et la taille réelle du fichier

Pourquoi ?

B) Pourquoi y a-t-il une différence entre les tailles données par les calculs et celles mesurées ?

	Taille mesurée	Taille calculée
Image en noir et blanc bitmap de 500x500	32062 octets	31250 octets
Image en noir et blanc bitmap de 400x400	20862 octets	20000 octets
Image en noir et blanc bitmap de 200x200	5662 octets	5000 octets

Comment le processeur « sait-il » que le fichier lu (« 00010101000101... ») est du texte, une image, du son, de la vidéo etc... ?

Un entête lui même codé en 0001...

<https://web.maths.unsw.edu.au/~lafaye/CCM/video/format-bmp.htm>

Peut indiquer →

C'est toujours en binaire

0001110101....

Que c'est une image

Le type d'image

La largeur de l'image en pixel

La hauteur de l'image en pixel

Le nombre de bits utilisés pour coder la couleur.

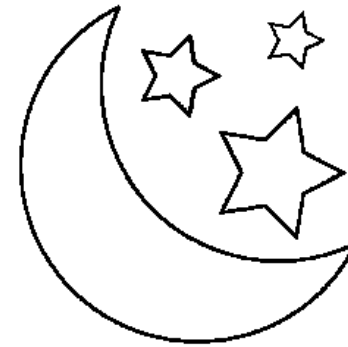
La méthode de compression

Les propriétés (titre, auteur etc...)

B) Pourquoi y a t-il une différence entre les tailles données par les calculs et celles mesurées ?

Exemple du fichier lunenoiretblanc.bmp

→ 32062 octets



0100101101110001110011

C'est une Image (ou
texte...etc...)

6496 bit
ou 812 octets

1111100011111

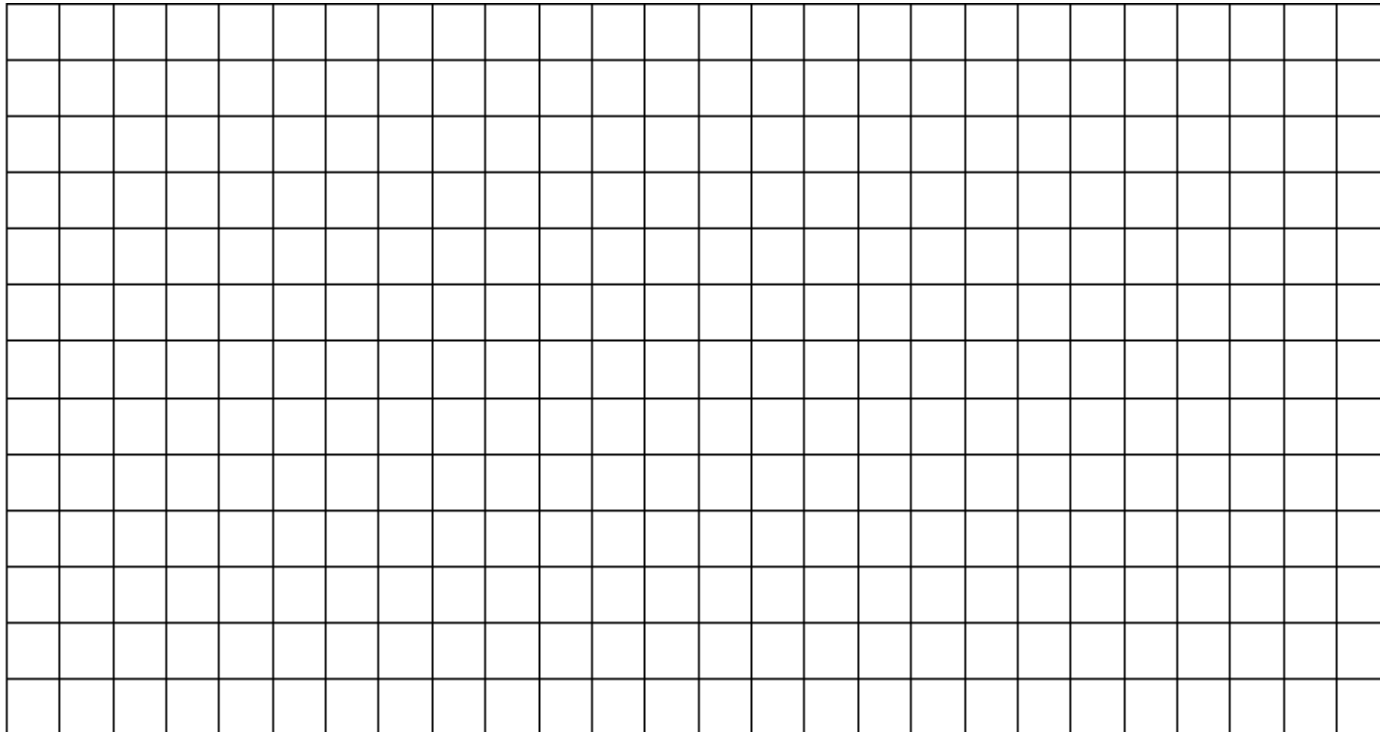
→ Codage de l'image
elle même
Pixel par pixel

256496 bit
Ou 31250 octets

Remarque un fichier sans en tête est par défaut considérée comme du texte...

En tête indiquant que c'est un BMP
monochrome (noir et blanc) De 26 X13 pixels.
puis....

```
1111111111111111111111111111001111111111001111111111000111110011111000  
11111100000111100001111000001110000000111000011100000001100000000000  
00000000000001000000000000000000000000000000000000000000000000000000  
100000000000000000000000000110000110001000010001100001110011111110011  
111111001111100111111100111111100111111111111111111111111111111111
```

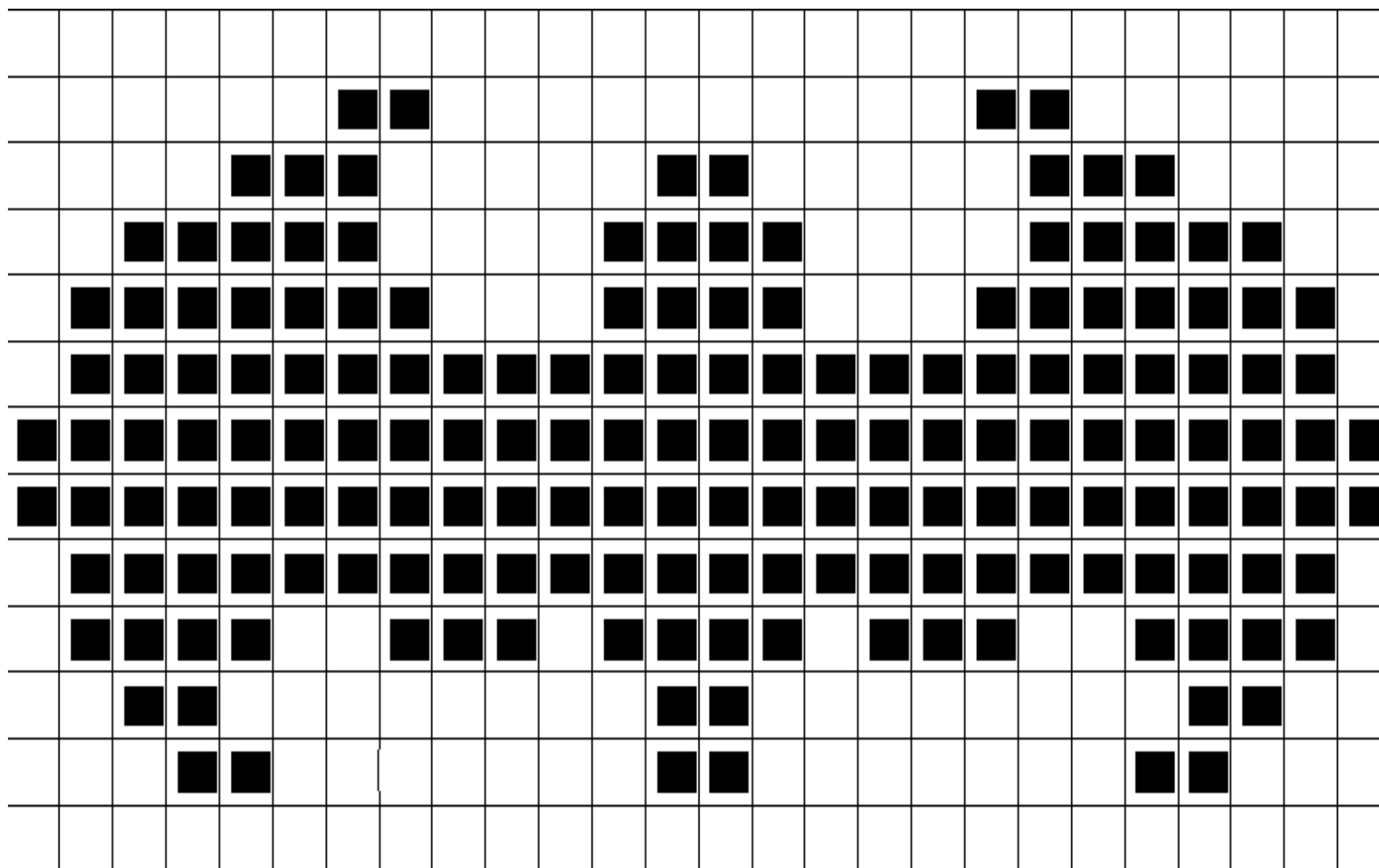


?

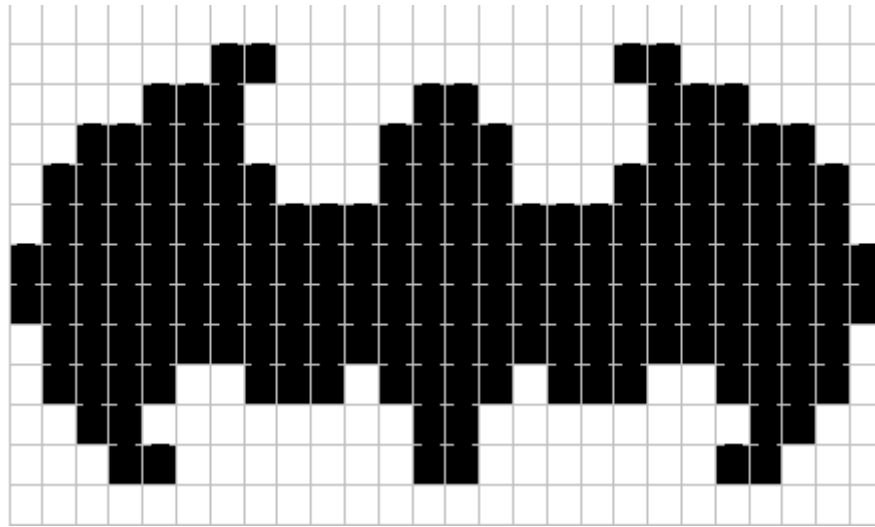
Indice



Réponse !



Remarque : Avant l'apparition des images « Ascii Art » : <http://www.ascii-fr.com/>



Questions :

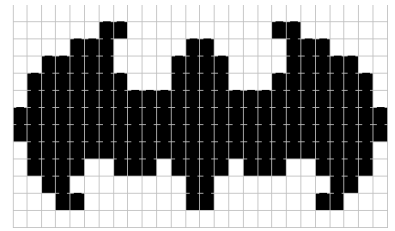
- 1) Combien de pixels constituent cette image ?
- 2) Quelle est la taille de ce codage en BIT ?
- 3) Quelle est la taille de ce codage en octet ?
- 4) voici le résultat de la mesure de la taille du fichier ?

Taille : 114 octets (114 octets)|

Questions :

1) Combien de pixels constituent cette image ?

Il y a 338 pixels (26x13)



2) Quelle est la taille de ce codage en BIT ?

En BMP monochrome : 1 pixel $\rightarrow$ 1 BIT

Ainsi le codage occupera 338 BITS

3) Quelle est la taille de ce codage en octet ?

1 octet = 8 BITS

Ainsi le codage occupera 43 Octet (42,25)

4) voici le résultat de la mesure de la taille du fichier ?

Taille : 114 octets (114 octets)|

Car mis a part le codage de l'image, un entête est codé en binaire indiquant que c'est une image BMP en noir et blanc , le nombre pixel en largeur et en hauteur et éventuellement d'autres informations (date,auteur...)

Plus petite unité d'information

binary Unit

Chiffre binaire

« Une lettre de l'ordinateur » : **1 Bit**

Une information numérique codée sur 16 Bit

0110101011000101

En Fait :

Unité informatique « usuelle »

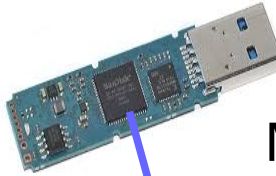
Octet = 8 bits

Une information numérique codée sur 2 octets

0110101011000101

Lien avec le stockage des fichiers

Périphérique de stockage

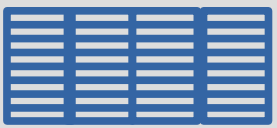


disque dur
Clef USB
Mémoire vive

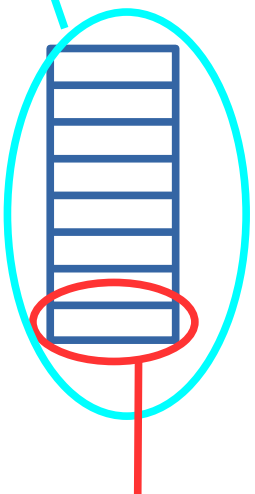
unité de mémoire : **Octet** = 8 bits
C'est historique

Dans 4 Go =
4000000000 octets
= 32000000000 bit

Clef USB



Des milliards de « cases »



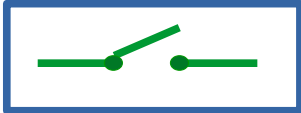
Vocabulaire :
1 case : 1 bit

Dans un bit soit
1 soit 0

Comme un interrupteur



1



0

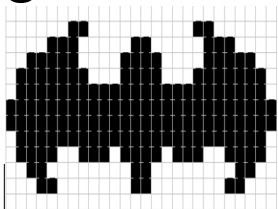
- Soit fermé :
« courant électrique » ~1

Soit ouvert : « pas de
courant électrique » ~ 0

Enregistrer un fichier c'est comme ouvrir ou fermer des interrupteurs des différents « bits » qui composent l'unité de stockage.

Cf : Petite anatomie d'unités de stockage

Enregistrement d'une image monochrome



1100111111111100011111001111100011111
10000011110000111100000111000000011
100001110000000.....

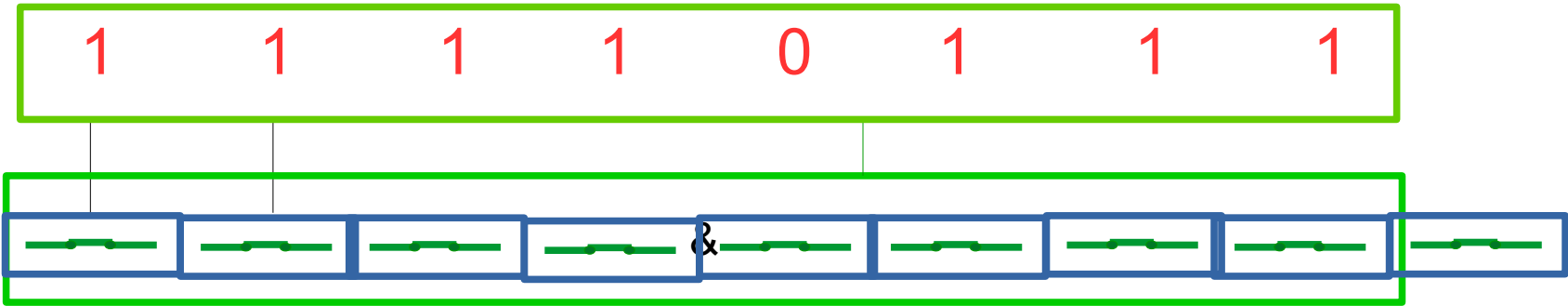
En tête

Traduction en
binaire

1111111111111111111111111111111110011111
1111100111111111111100011111001111100011
11110000011110000111100000111000000
0111000011100000001.....

Les pixels

Début de l'enregistrement



1 pixel = 1 BIT

8 pixel = 8 BIT = 1 Octet



Capacité clef USB

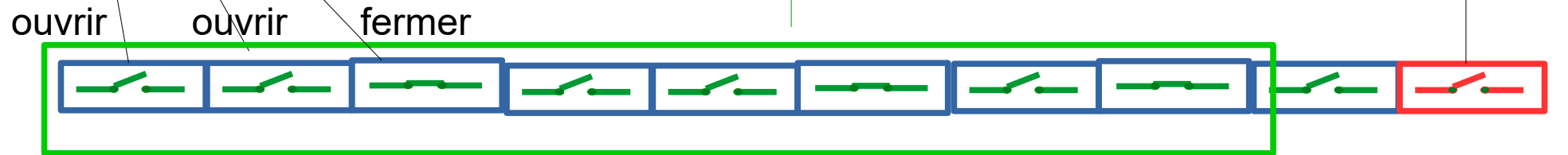
4 Go = 4000000000 octets = 32000000000 bit

Enregistrer un fichier informatique

001001010100101011100011111...

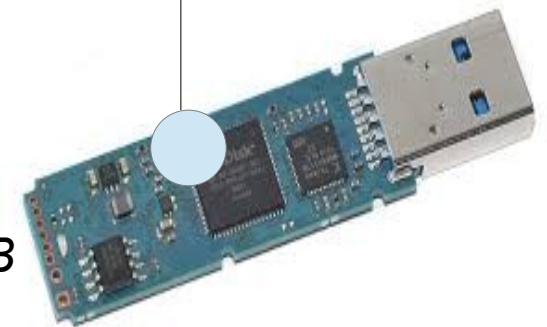
8 BIT = 1 Octet

1 BIT



X 1000000000

En fait, il n'y a pas vraiment d'interrupteurs, comme ceux que l'on voit sur les murs.. il s'agit d'une image pour comprendre le principe d'enregistrement des fichiers...



Cf : Petite anatomie d'unités de stockage

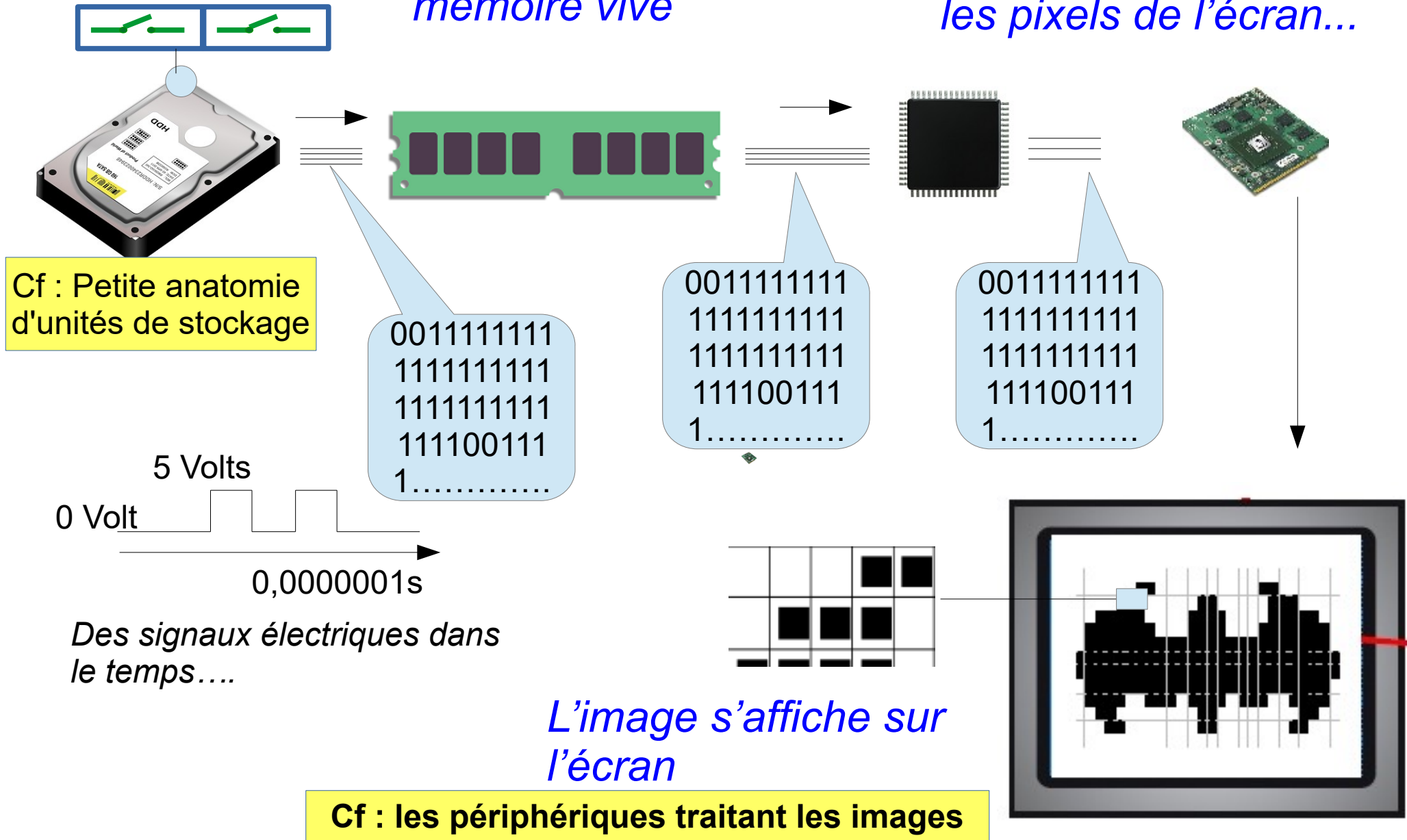
Capacité clef USB

4 Go = 4000000000 octets
= 32000000000 bit

Fichier enregistré dans le disque dur

Les données (codage binaire de l'image) sont chargées dans la mémoire vive

Le processeur traite les données et envoie les signaux pour allumer les pixels de l'écran...



suite

Comment coder (enregistrer) une image couleur ?

