Données 1 : **Binaire, octet et stockage**

I – Problème réel : magie magie

II – La base 10 (le décimal)

2.1 - Codification des nombres

La **case de poids faible** est **la plus à droite**. Elle code la valeur ________ qu'on peut écrire ____________

A chaque fois qu'on se déplace vers la gauche, la case code _____ fois plus que la précédente.

	Millier	Centaine	Dizaine	**Unité**
La case code	1000	100	10	**1**
La case code	10^3	10^2	10^1	$\mathbf{10^0}$
Nombre N	**5**	**2**	**4**	**3**
On obtient				

Au total, le nombre n vaut : N = ____

2.2 - Que peut-on mettre dans les cases en décimal ?

Les _____ **chiffres** de la base 10 sont : ____

On représente **un nombre** comme un **ensemble de chiffres**.

2.3 – Poids de case en puissance de 10 (voir le tableau ci-dessus)

- Unité $10^0 = 1$
- Dix (da pour déca) $10^1 = 10$
- Cent (h pour hecto) $10^2 = 10*10 = 100$
- Mille (**k** pour kilo) $10^3 = 10*10*10 = 1000$
- Million (**M** pour Mega) $10^6 = 10*10*10*10*10*10 = 1\ 000\ 000$
- Milliard (**G** pour Giga) $10^9 = 10*10*10*10*10*10*10*10*10 = 1\ 000\ 000\ 000$
- 1000 milliards (**T** pour Tera) $10^{12} = 1\ 000\ 000\ 000\ 000$

2.4 – Augmentation d'une unité (incrémentation)

Si le chiffre a un successeur, on le remplace par celui-ci

0 ⇨ 1 ⇨ 2 ⇨ 3 ⇨ 4 ⇨ 5 ⇨ 6 ⇨ 7 ⇨ 8 ⇨ 9

Sinon (pour 9), on change donc de procédé :
- on revient au premier chiffre (0) dans cette case ET
- on rajoute +1 dans la case juste à gauche (on rajoute une retenue).

Exercice de cours A : Trouver les successeurs des nombres suivants (en justifiant la réponse !)

Après **8** ? Après **9** ? Après **19** ? Après **13** ? Après **399** ?

2.3 – Nombre de cas dénombrables en base 10 (système décimal)

Avec X cases :

Valeurs différentes : $\mathbf{10^X}$

Valeur minimale : **0**

Valeur maximale : $\mathbf{10^X - 1}$

Exercice de cours B : Trouver le nombre de valeurs différentes disponibles et la valeur maximale avec 3 chiffres puis 6 chiffres en base 10.

III – La base 2 (le binaire)

3.1 - Que peut-on mettre dans les cases en binaire ?

Chaque case se nomme un **BIT**, mot qui provient de la contraction de **BINARY DIGIT**.

Les _____ **chiffres** de la base 2 sont : ____

3.2 - Codification des nombres

Le **bit de poids faible** est **le plus à droite**. Elle code la valeur ________ qu'on peut écrire ____________

A chaque fois qu'on se déplace vers la gauche, la case code _____ fois plus que la précédente.

DOC.1

				Unité
Le bit code				**1**
Le bit code	2^3	2^2	2^1	$\mathbf{2^0}$
Nombre N	**1**	**0**	**1**	**1**
On obtient				

Au total, le nombre n vaut : N = ____

Exercice 01 : Exprimer les **nombres binaires** suivant en base 10 :

N_A **= 1 0 0 0** N_B **= 0 0 1 0** N_C **= 0 1 1 0** N_D **= 1 1 1 0**

3.3 – Notation des bases :

On voit que c'est un peu étrange d'écrire quelque chose comme **1011 = 11** !
On indique la base utilisée par un nombre à l'aide d'un indice situé en bas à droite du nombre.
Exemple : $N = 1011_2 = 11_{10}$ ou $N = (1011)_2 = (11)_{10}$

3.4 – Incrémenter en binaire (rajouter 1)

- Un 0 devient son successeur : 1
- Un 1 étant le dernier chiffre en base 2, on repasse à 0 en rajoutant une retenue 1 dans la case juste à gauche.

Exercice de cours C : Trouver les nombres binaires suivants les exemples ci-dessous (en justifiant la réponse !) Après **0** ? Après **1** ? Après **10** ? Après **11** ? Après **1011** ?

Exercice 02 : Compléter la séquence suivante. Indiquer la valeur obtenue en base 10 dans la colonne de droite.

En base 2			En base 10
0	0	0	0
0	0	1	1
0	1	0	
1	1	1	7

3.5 – Nombre de cas dénombrables

Avec X cases : Valeurs différentes **:** $\mathbf{2^X}$ Valeur minimale **: 0** Valeur maximale : $\mathbf{2^X - 1}$

Exercice de cours D : Trouver le nombre de valeurs différentes disponibles et la valeur maximale avec 4 bits puis 8 bits puis 16 bits.

IV - Octet

4.1 - Octet : binaire vers décimal

Un octet est un ensemble de 8 bits successifs.

Puisqu'il contient 8 bits, un octet possède **2^8 (soit 256) valeurs différentes**, de **0 et 255**.
Sous forme binaire, un octet est donc compris entre **`0000 0000`** $_2$ et **`1111 1111`** $_2$.

DOC.2

Bit de poids fort							Bit de poids faible
128					4	2	1
2^7					2^2	2^1	2^0
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**

On peut alors écrire (**`1111 1111`**) $_2$ **=** ____

Exercice 03 : Justifier la valeur décimale correspondant à ces octets sous forme binaire :

- Octet A : `0100 1000`
- Octet B : `1000 0011`
- Octet C : `1111 1111`

Exercice 04 : En activant progressivement les bits de poids fort nécessaires, trouver l'encodage binaire des entiers suivants :

- Octet D : 37
- Octet E : 72
- Octet F : 123

Exercice 05 : Retrouver les résultats de l’exercice 04 mais en utilisant la division successive.

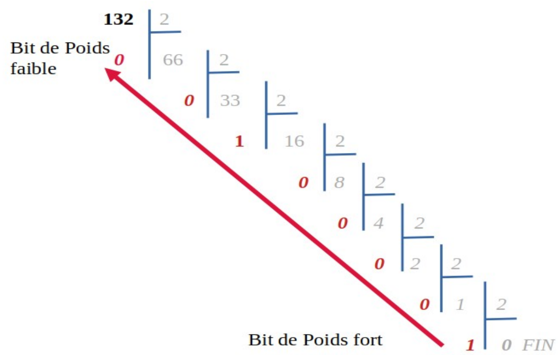
4.2 Activation progressive du bit de poids fort

On active progressivement les bits en commençant systématiquement par le bit le plus grand nécessaire pour encoder correctement le nombre entier.

4.3 Division successive par deux

La méthode de la division consiste à diviser le nombre par deux jusqu'à atteindre un quotient de 0. Les restes de la division euclidienne vont fournir les valeurs des bits.

132_{10} = 10000100_2



V – Utilisation de la notion d’octet

Un octet possède 256 valeurs différentes. Lorsqu'on veut mémoriser une information en machine, il suffit donc d'attribuer une valeur à une information. Cela se nomme un **encodage**. Les techniques d'encodage n'ont rien de secret : elles doivent êtres connue et diffusées de façon à pouvoir encoder puis à décoder.

5.1 – Encodage des entiers de 0 à 255

5.2 – Encodage de caractères

Les caractères sont stockés sous forme d’une suite d’octets.

Table d’encodage 1 octet : les 128 premiers caractères (0 à 127) sont communs à la plupart des encodages sur 1 octet : il s’agit de la **norme ASCII.** Les autres sont différents selon la table.

5.3 – Encodage des images

Chaque image est décomposée en pixels (un carré qui peut faire briller plus ou moins fortement).
En RGB, un **pixel est décomposé en 3 sous-pixels** : Rouge (Red), Vert (Green) et bleu (Blue).
On définit chaque luminosité par un nombre sur un octet (de 0 à 255).

255-0-0 sert à définir la couleur

0-255-0 sert à définir la couleur

175-0-0 sert à définir la couleur

255-255-0 sert à définir la couleur

VI – Sous-unités de l'octet

Sous-unité	Valeur en octet	Pourquoi ?
Un kilo (**k**) représente 10^3	1 000	
Un Méga (**M**) représente 10^6	1 000 000	un million, soit 1000 x 1 000
Un Giga (**G**) représente 10^9	1 000 000 000	un milliard, soit 1000 x 1 000 000
Un Téra (**T**) représente 10^{12}	1 000 000 000 000	mille millards
Un Péta (**P**) représente 10^{15}	1 000 000 000 000 000	Un million de milliards

Exercice 06° On considère un texte de 200 000 caractères. Calculer le nombre d'octets nécessaires à son stockage
en octets (o), en bits, en ko (ko) en Mo (Mo)

VII – Hexadécimal

7.1 - Codification des nombres

Les _____ **chiffres** de la base 16 sont :

On notera que $\mathbf{A}_{16} = \mathbf{10}_{10}$ $\mathbf{B}_{16} = \mathbf{11}_{10}$ $\mathbf{C}_{16} = \mathbf{12}_{10}$ $\mathbf{D}_{16} = \mathbf{13}_{10}$ $\mathbf{E}_{16 = 4}\mathbf{14}_{10}$ $F_{16} = \mathbf{15}_{10}$

On peut donc représenter un **nombre** comme un ensemble de cases contenant un chiffre.

La case de poids faible est la plus à droite. Elle code la valeur _________ qu'on peut écrire ______________
A chaque fois qu'on se déplace vers la gauche, la case code _____ fois plus que la précédente.

La case code	4096	256	16	1
La case code	16^3	16^2	16^1	16^0
Exemple N =	**5**	**2**	**D**	**A**
On obtient	5 x 4096	2 x 256	13 x 16	10 x 1

Au total, le nombre vaut : $\mathbf{N = 52DA}_{16} = (5*4096 + 2*256 + 13*16 + 10)_{10} = \mathbf{21210}_{10}$

Exercices 07/08/09 : Déterminer les valeurs suivantes en décimal : $\mathbf{41}_{16}$, $\mathbf{FF}_{16}$, $\mathbf{FE}_{16}$

7.2 – Augmentation d'une unité

Pour cela, il suffit de placer le prochain chiffre dans la liste
0 ⇨ 1 ⇨ 2 ⇨ 3 ⇨ 4 ⇨ 5 ⇨ 6 ⇨ 7 ⇨ 8 ⇨ 9 ⇨ A ⇨ B ⇨ C ⇨ D ⇨ E⇨ F

- Si on est déjà au dernier chiffre (F), on revient au premier chiffre (0) dans cette case ET on rajoute UN dans la case juste à gauche.

Exercice de cours : Quel est le nombre suivant **en base 16 ?** On travaillera avec

E + 1 = F + 1 = 13 + 1 = F3 + 1 = FD + 1 =

FF + 1 = 3F + 1 =

7.3 – Nombre de cas dénombrables

Si on dispose de X cases pour écrire un nombre en base 10, on dispose de $\mathbf{16^X}$ valeurs possibles.

Attention, la valeur minimale étant le cas 0, la valeur maximale est $\mathbf{16^X - 1}$

Exercice de cours : Trouver le nombre de cas et la valeur maximale avec 2 chiffres ou 3 chiffres en base 16.

DOCUMENTS Données 1 : **Binaire, octet et stockage**

III – La base 2 (le binaire)

DOC.1

				Unité
Le bit code				**1**
Le bit code	2^3	2^2	2^1	$\mathbf{2^0}$
Nombre N	**1**	**0**	**1**	**1**
On obtient				

Au total, le nombre n vaut : N = ____

IV – OCTET

DOC.2

Bit de poids fort — Bit de poids faible

128					4	2	1
2^7					2^2	2^1	2^0
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**

On peut alors écrire (**1111 1111**)$_2$ = ____

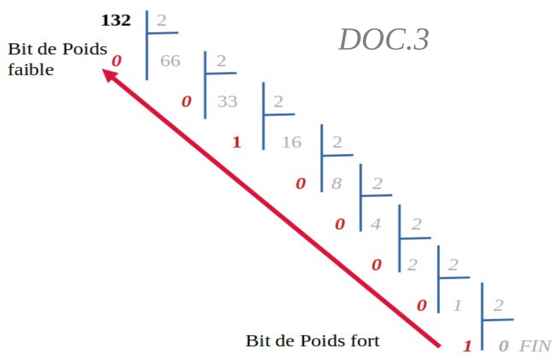
4.2 Activation progressive du bit de poids fort

On active progressivement les bits en commençant systématiquement par le bit le plus grand nécessaire pour encoder correctement le nombre entier.

4.3 Division successive par deux

La méthode de la division consiste à diviser le nombre par deux jusqu'à atteindre un quotient de 0. Les restes de la division euclidienne vont fournir les valeurs des bits.

DOC.3



EXERCICES

Exercice 01 : Exprimer les **nombres binaires** suivant en base 10 :

N_A = 1 0 0 0 **N_B = 0 0 1 0**
N_C = 0 1 1 0 **N_D = 1 1 1 0**

Exercice 02 : Compléter la séquence. Indiquer la valeur obtenue en base 10.

En base 2

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

En base 10

0
1
7

Exercice 03 : Justifier la valeur décimale correspondant à ces octets sous forme binaire :

- Octet A : `0100 1000`
- Octet B : `1000 0011`
- Octet C : `1111 1111`

Exercice 04 : En activant progressivement les bits de poids fort nécessaires, trouver l'encodage binaire des entiers suivants :

- Octet D : 37
- Octet E : 72
- Octet F : 123

Exercice 05 : Retrouver les résultats de l'exercice 04 mais en utilisant la division successive.