

**QCM - Mesure de l'information, codage du son, des images et des vidéos**

**Q1** - Calculez en bits les valeurs suivantes

- 10 octets = **10x8 bits**

Je rappelle qu'un octet vaut 8 bits

- 2 Méga octets (ou 2 Mo) = **2x10<sup>6</sup> octets**  
= **2x10<sup>6</sup> x 8 bits = 16 millions de bits**

Je rappelle qu'un Méga vaut 10<sup>6</sup>

- 2 Kibi octets (ou 2 KiO) = **2 x 2<sup>10</sup> x 8 bits**  
= **16 x 1024 bits = 16384 bits**

Je rappelle qu'un kibi vaut 2<sup>10</sup>

- 1 Tera octets (ou 1 To) = **10<sup>12</sup> octets**  
= **8 x 10<sup>12</sup> bits**

je rappelle :

1kilo = 10<sup>3</sup>  
1Méga = 10<sup>6</sup>  
1Giga = 10<sup>9</sup>  
1Téra = 10<sup>12</sup>

- 1 Gibi octets (ou 1 GiO) = **2<sup>30</sup> octets**  
= **8 x 2<sup>30</sup> bits**

je rappelle :

1kibi = 2<sup>10</sup>  
1Mébi = 2<sup>20</sup>  
1Gibi = 2<sup>30</sup>  
1Tébi = 2<sup>40</sup>

**Q2** - Le son (musique par exemple) est par définition

- un signal analogique  
 un signal numérique  
 est une vibration mécanique d'un fluide (de l'air notamment), qui se propage sous forme d'ondes

**Q3** - Un son numérique est caractérisé par :

- sa **fréquence** : Le nombre de « relevés » effectués chaque seconde, elle est mesuré en Hertz
- sa **résolution numérique** : le nombre de bits permettant de coder chaque valeur de l'amplitude relevée

Le calcul de la capacité mémoire d'un fichier son non compressé est comme suit :

poids (octet) = Fréquence d'échantillonnage (Hz) x Résolution (octet) x Durée (seconde)

Si j'ai effectué un enregistrement sonore pendant une minute d'un son avec une fréquence de 1khz et une résolution de 8 bits, Calculez la capacité mémoire du fichier généré

**Réponse :**

- J'ai une durée d'une minute= 60 secondes
- avec une fréquence de 1khz je capture 1000 valeurs du son en une seconde
- Pour chaque valeur j'utilise 8 bits

**ce qui me donne au total : 1000 x 8 x 60 = 480.000 bits**

Si j'augmente la fréquence de mon enregistrement comment sera la qualité du fichier son généré sera :

- plus faible**    **meilleure**

**Augmenter la fréquence veut dire capturer plus de valeur du signal sonore (qui est analogique par nature). Cela veut qu'on va se rapprocher du signal original. N'oubliez pas que dès que vous stocker un son dans votre ordinateur, c'est une approximation de la réalité que vous avez ! Plus vous prenez de valeurs, plus le son représenté dans votre ordinateur est fidèle à l'original.**

Si j'augmente la résolution numérique de mon enregistrement au lieu de 8 bits, je choisis 16 bits la qualité du fichier son généré sera :

- plus faible**    **meilleure**

**Augmenter la résolution numérique du son c'est utiliser plus de bits pour représenter l'amplitude du signal. Là aussi nous aurons un fichier son plus fidèle à l'original.**

**Dans cette question, vous devez comprendre que le son que vous stocker sur votre ordinateur n'est pas identique au son généré dans la réalité comme celui de votre voix ou celui d'une guitare. C'est une représentation approximative de la réalité que vous représenté en capturant des valeurs de votre son par unité de temps. Plus vous augmentez la fréquence de cette capture, plus vous ne ferez pas d'erreur en fréquence du son, plus vous augmentez le nombre de bit de la résolution du son, plus vous ferez moins d'erreur dans le volume du son.**

**Q4** – Complétez le tableau suivant :

Format	Compression
wave	<input checked="" type="checkbox"/> Oui <input type="checkbox"/> Non
mp3	<input type="checkbox"/> Oui <input checked="" type="checkbox"/> Non
ogg	<input type="checkbox"/> Oui <input checked="" type="checkbox"/> Non
RAW	<input checked="" type="checkbox"/> Oui <input type="checkbox"/> Non
AAC	<input type="checkbox"/> Oui <input checked="" type="checkbox"/> Non

**Je rappelle que la compression est une opération consistant à réduire le nombre de bits utilisés pour coder le son, l'image ou la vidéo tout en essayant de ne pas perdre en qualité.**

**Q5** – On a créé des codages du son, de l'image et de la vidéo compressés au lieu de les garder bruts (sans compression) ?

- Juste pour s'amuser
- pour réduire la taille des fichiers
- pour rendre rapide les programmes
- pour assurer des transferts plus rapides

**Commentaires :** si vous compressez des fichiers, vous aurez des capacités plus petites donc moins de stockage et est plus rapide de les transférer. Par exemple, les images sur le Web sont en général au format JPEG ou PNG, le son est le plus souvent du MP3 et la vidéo du MP4, ces formats permettent d'avoir des fichiers de tailles réduites permettant ainsi de les transférer même si vous avez un faible débit d'Internet.

**Q6** – Pour calculer la définition d'une image, on utilise deux valeurs :

- la surface (en pixels) et la largeur (en pixels)
- la longueur (en pixel) et la largeur (en pixels)
- la profondeur (en bits) et la largeur (pixels)

**Q7** – La profondeur d'une image définit l'ensemble de ses couleurs. Elle est mesurée en bits. Une profondeur de 4 bits permet d'avoir  $2^4$  couleurs différentes. Combien de bits sont utilisés pour la profondeur des images en codage RVB « True color » ? **24 bits**

**Commentaire :** RVB veut dire codage selon les 3 couleurs de base Rouge, Vert et Bleu. En True-type, on utilise 8 bits pour chaque couleur, ce qui donne  $8 \times 3 = 24$  bits.

**Q8** – Lorsqu'on vous dit qu'une image est codée en RVB que veut dire ces lettres :

- R : Rouge
- V : Vert
- B : Bleu

**Q9** : En supposant que vous codez en True color (RVB), indiquez les couleurs représentées par les codes suivants :

$(255, 0, 0)_{10}$	Rouge maxi (Vif)
$(0, 0, 255)_{10}$	Vert maximum (vif)
$(0, 255, 255)_{10}$	Vert + Bleu = Cyan
$(5, 2, 5)_{10}$	RVB proche de 0 donc NOIR
$(5, 0, 255)_{10}$	Rouge et Vert proches de 0 et vert maxi donc VERT
$(0, 255, 255)_{10}$	Vert + Bleu = Cyan
$(0, 0, 0)_{10}$	NOIR
$(50, 50, 50)_{10}$	Gris <b>foncé</b> car les couleurs RVB sont proches de 0 donc du noir
$(200, 200, 200)_{10}$	Gris <b>clair</b> car les couleurs RVB sont proches de 255 donc du blanc

$R+B = magenta$   $R+V = Jaune$   $B+V = Cyan$   
 $R+V+B$  avec le même dosage donne du gris. Proche de 0 c'est foncé et proche de 255 c'est clair !

**Q10** – En supposant que vous avez une définition d'une image 200x300, calculez le poids (capacité mémoire) de cette image si sa profondeur est codée :

- en « True type » RVB (24 bits) :  $200 \times 300 \times 24 = 1.440.000$  bits
- sur 8 bits (256 couleurs) :  $200 \times 300 \times 8 = 480.000$  bits
- en noir et blanc :  $200 \times 300 \times 1 = 60.000$  bits

Je rappelle que le poids d'une image est égale à sa résolution x sa profondeur = nombre de ligne x nombre de colonnes x nombre de bits codant la couleur.

**Q11** – On considère qu'on a une animation lorsqu'on fait défiler au minimum  $n$  images par secondes. Indiquez combien vaut ce nombre  $n$  ? En général, une dizaine d'images par seconde donne un effet d'animation

Je rappelle que c'est l'imperfection de notre vision qui est exploitée dans la génération d'animation. En général, notre cerveau n'est pas capable de distinguer 2 images en moins de un vingt-cinquième (1/25) de seconde

**Q12** – Lorsqu'on code une vidéo, il est souhaitable de faire défiler le maximum d'images par secondes au moins 100 images par seconde pour avoir une très bonne qualité de l'animation ?

- Vrai
- Faux

**Commentaire :** C'est tout à fait faux ce qui est déclaré ci-dessus. En effet, comme déjà mentionnée, notre cerveau n'est pas capable de distinguer 2 images en moins de un vingt-cinquième (1/25) de seconde. Au-delà de 25 images par secondes, la qualité de l'image est très faiblement améliorée (du moins on la perçoit pas!). D'ailleurs, le standard du codage de la télévision en Europe et chez nous est de 25 images par seconde !

**Q13 – Codage des entiers :** Donnez l'étendue des valeurs du codage S+VA, C1 et C2:

Codages	Étendue des valeurs si le codage est sur	
	n bits (bit de signe compris)	5 bits (bit de signe compris)
S+VA	$[-(2^{n-1} - 1), + (2^{n-1} - 1)]$	$[-15, + 15]$
C1	$[-(2^{n-1} - 1), + (2^{n-1} - 1)]$	$[-15, + 15]$
C2	$[-2^{n-1}, + (2^{n-1} - 1)]$	$[-16, + 15]$
Binaire pure (entiers non signés)	$[0, 2^n - 1]$	$[0, 31]$

**Q14 – Codage S+VA, C1 et C2 avantage et inconvénients**

Complétez le tableau suivant :

	Avantages	Inconvénients
S+VA	-Représentation naturelle et simple.	-Calculs non évidents. -Double représentation du zéro.
C1	-Représentation des nombres négatifs relativement simple.  - Calculs possibles.	-Double représentation du zéro.  -Problème de performance du à l'addition de la retenus.
C2	-Représentation des nombres négatifs plus compliquée. -Calculs possibles.	-Meilleure performance (on additionne pas la retenue).

**Q15 –** Donnez sur **8 bits**, en S+VA, C1 et C2 le codage des nombres  $(43)_{10}$  et  $(-43)_{10}$

Codage de $(43)_{10}$	
S+VA	<b>(00101011)</b>
C1	<b>(00101011)</b> (+43) positif. Il est représenté de la même manière qu'en codage (S+VA).
C2	<b>(00101011)</b> (+43) positif. Il est représenté de la même manière qu'en codage (S+VA).

Codage de $(-43)_{10}$	
S+VA	<b>(1 0101011)</b> Il suffit de changer juste le bit de signe.
C1	<b>(1 1010100)</b> Il suffit d'inverser tous les bits de la valeur absolue.
C2	<b>(1 1010101)</b> Il suffit de prendre la valeur absolue, de recopier tous les 0 en commençant de la droite, arrivé au premier 1, il faut le recopier, puis inverser le reste.

**Q16 – Calcul arithmétique :**

Effectuez les calculs suivants

- $15 + (-63)$  en C1 sur 8 bits

$$(15)_{10} = (00001111)_{SVA} = (00001111)_{C1}$$

$$(-63)_{10} = (10111111)_{SVA} = (11000000)_{C1}$$

<b>15</b>	0	0	0	0	1	1	1	1
<b>-63</b>	1	1	0	0	0	0	0	0
<b>-48</b>	1	1	0	0	1	1	1	1

- $13 + (-65)$  en C2 sur 8 bits

$$(13)_{10} = (00001101)_{SVA} = (00001101)_{C2}$$

$$(-65)_{10} = (10111111)_{SVA} = (11000001)_{C2}$$

<b>13</b>	0	0	0	0	1	1	0	1
<b>-63</b>	1	1	0	0	0	0	0	1
<b>-50</b>	1	1	0	0	1	1	1	0

- $73 + 86$  en C2 sur 7 bits

$$(73)_{10} = (1001001)_{SVA} = (1001001)_{C2}$$

$$(86)_{10} = (1010110)_{SVA} = (1010110)_{C2}$$

<b>73</b>	1	0	0	1	0	0	1
<b>86</b>	1	0	1	0	1	1	0
<b>159</b>	0	0	1	1	1	1	1

Indiquez le calcul qui provoque un débordement.

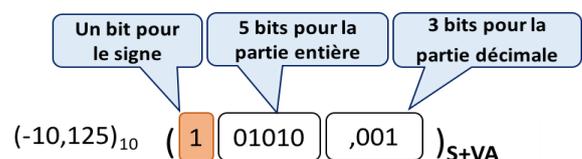
Le calcul qui provoque un débordement est le troisième calcul  $(73 + 86)$ . D'ailleurs, vérifions l'étendue des valeurs que l'on peut représenter sur 7 bits en C2 :

$$[-2^{7-1}, 2^{7-1}-1] = [-64, +63].$$

En faisant le calcul de 73 et de 86 on obtient 159 ce qui est en dehors de l'intervalle des valeurs possibles en C2 sur 7 bits !

**Q17 – Codage en virgule fixe :**

Sur **9 bits** dont un bit de signe et 5 bits pour la partie entière et 3 bits pour la partie décimale, donnez les représentations des nombres suivants :



$$(-12,125)_{10} = (1\ 10011,110)_{c1}$$

$$(-12,125)_{10} = (1\ 10011,111)_{c2}$$

### Q18 – Codage des caractères :

En vous référant à la table ASCII standard sur 7 bits  
donnez le code (en binaire) du mot « MP3 »

	000	001	010	011	100	101	110	111
0000	NULL	DLE	0	@	P	`	p	
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EDT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

Exemple le code de A est  $(1000001)_2 = (65)_{10} = (41)_{16}$

	000	001	010	011	100	101	110	111
0000	NULL	DLE	0	@	P	`	p	
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EDT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

$(P)_{ASCII} = (1010000)_2$   
 $(3)_{ASCII} = (0110011)_2$   
 $(M)_{ASCII} = (1001101)_2$

Ce qui donne :

$$(MP3)_{ASCII} = (1001101\ 1010000\ 0110011)$$