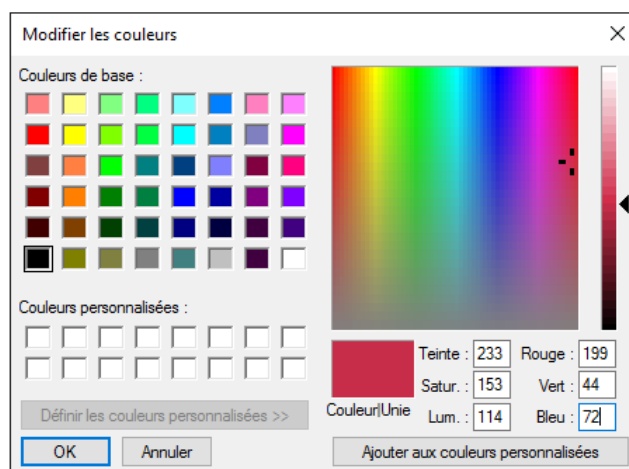


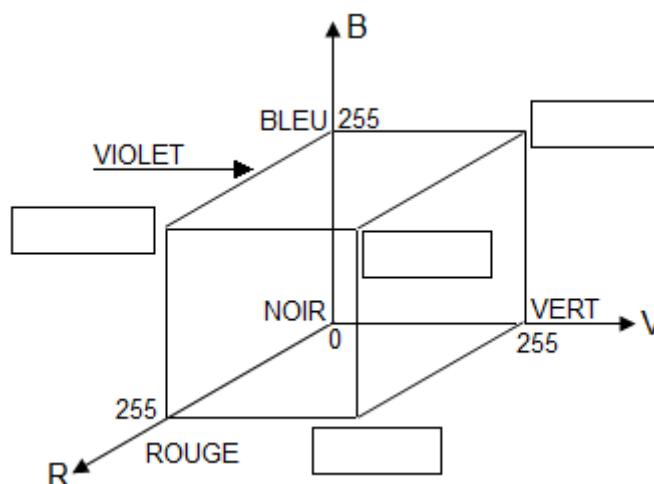
## IMAGES NUMÉRIQUES ET CODAGE RVB DES COULEURS



### 1 / Codage RVB des couleurs

CONSULTER l'introduction et la ressource « Images Bitmap et codage RVB des couleurs ».



**Q1 / COMPLÉTER** la figure ci-dessous : **INDIQUER** les couleurs correspondant aux différents sommets du cube.



**Q2 / DÉTERMINER** combien de couleurs sont disponibles au total.

**Q3 / COMPLÉTER** le tableau suivant :

Composantes R V B <sub>(10)</sub>	Composantes R V B <sub>(16)</sub>	Couleur	Nom
255 127 39			
	80 6D 5A		Châtaigne
			Violet

	BE F5 74		Pistache
96 96 96			

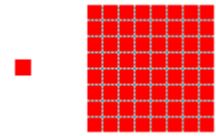
Remarque : Le violet est un mélange de Magenta (255 ; 0 ; 255) et de Bleu (0 ; 0 ; 255) en proportions égales.

Conseil : **S'AIDER** de la calculatrice « *Windows* » (mode scientifique ou programmeur selon version).

## 2 / Analyse de quelques images simples

**CONSULTER** la ressource « Organisation d'un fichier BMP ».

On considère l'image ci-contre (avec une vue au zoom 800% sur laquelle un quadrillage a été rajouté de manière à distinguer les pixels). Elle consiste en un carré de couleur rouge uniforme.



**Q4 / CALCULER** la taille (nombre d'octets) du contenu bmp. **INDIQUER**, toujours en octets, la taille totale du fichier « red.bmp ». Pour cela, **PLACER** le pointeur de souris sur le fichier correspondant dans l'explorateur de fichiers *Windows*.

**Q5 / EN DÉDUIRE :**

- la taille de l'en-tête du fichier bitmap :
- l'adresse de décalage (adresse de début du contenu bmp au sein du fichier) :

Remarque : la première adresse est  $00_{(10)} = 00_{(16)}$ .

On donne ci-dessous une image (avec le même zoom que précédemment) et le contenu correspondant du fichier bmp, analysé à l'aide du logiciel « *EditHexa* ».

**Q6 / ENTOURER :**

- l'en-tête du fichier (en bleu) ;
- le contenu bmp (en vert).

**DÉCRIRE** l'image.

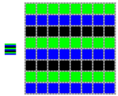


Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00000000	42	4D	F6	00	00	00	00	00	00	00	36	00	00	00	28	00
00000010	00	00	08	00	00	00	08	00	00	00	01	00	18	00	00	00
00000020	00	00	C0	00	00	00	C4	0E	00	00	C4	0E	00	00	00	00
00000030	00	00	00	00	00	00	48	2C	C7	48	2C	C7	48	2C	C7	48
00000040	2C	C7	48	2C	C7	48	2C	C7	48	2C	C7	48	2C	C7	48	2C
00000050	C7	48	2C	C7	48	2C	C7	48	2C	C7	48	2C	C7	48	2C	C7
00000060	48	2C	C7	48	2C	C7	48	2C	C7	48	2C	C7	48	2C	C7	48
00000070	2C	C7	48	2C	C7	48	2C	C7	48	2C	C7	48	2C	C7	48	2C
00000080	C7	48	2C	C7	48	2C	C7	48	2C	C7	48	2C	C7	48	2C	C7
00000090	48	2C	C7	48	2C	C7	48	2C	C7	48	2C	C7	48	2C	C7	48
000000A0	2C	C7	48	2C	C7	48	2C	C7	48	2C	C7	48	2C	C7	48	2C
000000B0	C7	48	2C	C7	48	2C	C7	48	2C	C7	48	2C	C7	48	2C	C7
000000C0	48	2C	C7	48	2C	C7	48	2C	C7	48	2C	C7	48	2C	C7	48
000000D0	2C	C7	48	2C	C7	48	2C	C7	48	2C	C7	48	2C	C7	48	2C
000000E0	C7	48	2C	C7	48	2C	C7	48	2C	C7	48	2C	C7	48	2C	C7
000000F0	48	2C	C7	48	2C	C7										

**Q7 / ANALYSER** l'en-tête puis **COMPLÉTER** le tableau suivant.

	Adresse	Hexadécimal	ASCII	Décimal
Nombre magique	00			
Taille fichier	02			
Décalage	0A			

On donne ci-dessous une image (avec un zoom) et l'en-tête du fichier bmp correspondant, toujours obtenu avec « EditHexa ».



Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00000000	42	4D	F6	00	00	00	00	00	00	00	36	00	00	00	28	00
00000010	00	00	08	00	00	00	08	00	00	00	01	00	18	00	00	00
00000020	00	00	C0	00	00	00	00	00	00	00	00	00	00	00	00	00
00000030	00	00	00	00	00	00										

**Q8 / COMPLÉTER** le contenu bmp correspondant aux trois premières lignes de l'image : **ÉCRIRE** le code hexadécimal de la couleur du pixel correspondant.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
30	00	00	00	00	00	00										
40																
50																
60																
70																

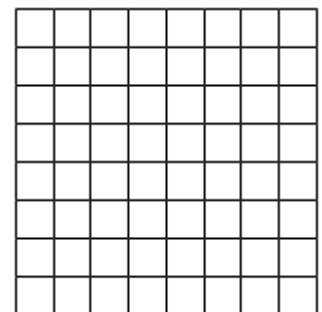
On donne ci-dessous une portion du fichier correspondant à une image de 8 x 8 pixels.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
30							FF	FF	FF	00	00	00	FF	FF	FF	00
40	00	00	FF	FF	FF	00	00	00	FF	FF	FF	00	00	00	00	00
50	00	FF	FF	FF	00	00	00	FF	FF	FF	00	00	00	FF	FF	FF
60	00	00	00	FF	FF	FF	FF	FF	FF	00	00	00	FF	FF	FF	00

La séquence bmp étant répétitive, seul le début est indiqué.

**Q9 / COLORIER** ci-contre l'image correspondante. **DÉCRIRE** le résultat obtenu.

Remarque : Le quadrillage permet de distinguer les pixels.



**LANCER** l'application « EditHexa » et **OUVRI**R le fichier « Green.bmp » (VOIR image ci-contre).



On souhaite modifier l'en-tête de ce fichier pour y incorporer la signature « SIN » (caractères ASCII majuscules « S », « I » et « N »).

4 octets sont disponibles à partir de l'adresse 06<sub>(16)</sub>.

**Q10 / ETABLIR** le contenu correspondant en hexadécimal.

**EFFECTUER** la modification sous « *EditHexa* » puis **VÉRIFIER** sous « *Paint* » que l'image n'est pas altérée.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00																

Remarque : Le logiciel « EditHexa » permet d'accéder à la correspondance entre les caractères et le nombre correspondant (en décimal, hexadécimal, binaire) grâce au menu Outils \ Table des caractères.

Validation professeur

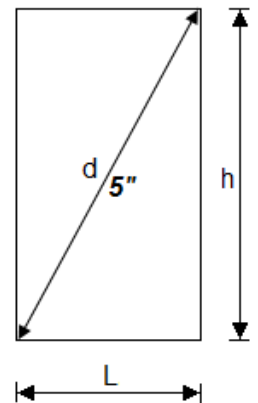
### 3 / Taille écran, résolution, espace mémoire

Un smartphone présente les performances suivantes, annoncées par son constructeur :

- écran 5 " au format 16/9, 1280 x 720 px
- appareil photo 8Mpixels (mode normal ; 2Mpixels en selfie)

Rappel : 1" (1 pouce) = 2,54cm

**Q11 / MONTRER** que les dimensions de l'écran « **h** » et « **L** » valent respectivement 111 et 62mm.



**Q12 / VÉRIFIER** que la résolution « **r** » de l'écran est proche de 300 ppp (ou dpi) quelle que soit la direction.

Remarque : ppp  $\equiv$  point par pouce (en anglais dpi  $\equiv$  dot per inch).

**Q13 / CALCULER** le nombre de pixels d'une photo prise en mode normal avec cet appareil en :

- vertical (une colonne) : « **Nv** »
- horizontal (une ligne) : « **Nh** »

**VÉRIFIER** que la taille de l'image est conforme aux performances annoncées par le constructeur.

**Q14 / EN DÉDUIRE** l'espace mémoire occupé par une photo et **DÉTERMINER** le nombre de photos qu'il est théoriquement possible de stocker sur la carte SD additionnelle, d'une capacité de 16Go.

Remarque : **NE PAS TENIR COMPTE** de l'en-tête du fichier au format « *BMP* » généré à chaque photographie.

Fondamental :

1ko = 1 024 octets =  $2^{10}$  octets

1Mo = 1 048 576 octets =  $2^{20}$  octets

1Go = 1 073 741 824 octets =  $2^{30}$  octets

**Q15 / JUSTIFIER** que les fonctionnalités du smartphone permettent de zoomer dans l'image sans que celle-ci paraisse trop pixelisée.

#### 4 / Autres formats d'images

**Q16 / INDIQUER** les avantages et les inconvénients des images au format bmp.

**Q17 / PROPOSER** une solution pour répondre au principal inconvénient des images bmp.

**Q18 / A partir d'une rapide recherche internet, CITER d'autres formats d'images et INDIQUER** succinctement leur intérêt.

## Pour aller plus loin



**DÉTERMINER** (explorateur de fichiers Windows) la taille (en kilo octets) de l'image « BLUE.bmp ».

**OUVRIR** le fichier sous « *Paint* » puis l'**ENREGISTRER** au format « png ».

**DÉTERMINER** la taille (en octets) du nouveau fichier « BLUE.png », **VÉRIFIER** :

- la compression ;
- l'altération éventuelle de l'image.

**EXAMINER** les deux fichiers sous « *EditHexa* » et **COMPLÉTER** le tableau ci-dessous.

Nom du fichier	Type (nombre magique)			
	Nombre octets	Adresse début	Caractères ASCII	Hexa
BLUE.bmp			BM	
BLUE.png			PNG	

	NT	0	1	2	3
Préparation					
1 / Codage RVB des couleurs					
2 / Analyse de quelques images simples					
3 / Taille écran, résolution, espace mémoire					
4 / Autres formats d'images					

*NT : Non Traité – 0 : Incorrect ou très insuffisant – 1 : Le négatif l'emporte sur le positif  
2 : Le positif l'emporte sur le négatif – 3 : Correct*