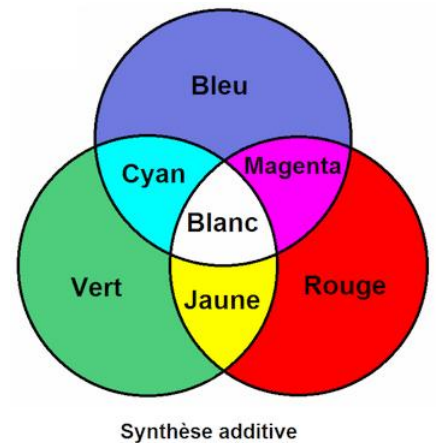


Spécialité NSI Première	DEVOIR SURVEILLE DE	Jeudi 2 décembre 2021
Lycée d'Avesnières		Durée : 55 mn
Année scolaire 2021-2022		Calculatrice interdite

Exercice 1 (6 points)

Sur un écran d'ordinateur ou de smartphone, les couleurs sont créées en mélangeant du rouge, du vert, du bleu. C'est la synthèse additive des couleurs. On imagine un dispositif dans lequel trois lampes de chacune de ces couleurs sont dirigées vers le même endroit et peuvent être allumées ou éteintes.

Une couleur du dispositif sera par exemple codée par le code 110. Cela signifie que les lampes rouges et vertes sont allumées et la bleue est éteinte.



1) Justifier que ce dispositif n'est pas capable de produire plus de huit couleurs différentes.

2) Le tableau ci-contre donne les huit couleurs et leurs codes :

Le complément d'une couleur est obtenu de la façon suivante :

- Si une lampe est allumée alors on l'éteint.
- Si une lampe est éteinte alors on l'allume.

Quel est le complément de chacune des huit couleurs figurant dans le tableau ?

Couleur	Rouge	Vert	Bleu
Noir	0	0	0
Bleu	0	0	1
Vert	0	1	0
Cyan	0	1	1
Rouge	1	0	0
Magenta	1	0	1
Jaune	1	1	0
Blanc	1	1	1

3) Il est également possible de faire une opération logique *bit à bit* entre deux codes. Le tableau ci-contre montre le fonctionnement de l'opérateur or noté |, de l'opérateur and noté & et de l'opérateur xor noté ^.

Donner la couleur obtenue si on fait les opérations suivantes :

- bleu | rouge
- magenta & cyan
- vert ^ blanc

```

a = 0b1100
b = 0b1010

c = a | b
d = a & b
e = a ^ b

>>> bin(c)
'0b1110'
>>> bin(d)
'0b1000'
>>> bin(e)
'0b0110'

```

Exercice 2 (5 points)

La table ci-après donne le code associé à chacun des caractères ASCII imprimables. Les cases vides correspondent à des caractères non imprimables comme des caractères de contrôle.

Le code du caractère en hexadécimal s'obtient en écrivant le numéro de la ligne suivi du numéro de la colonne ; par exemple, la lettre *M* a pour code hexadécimal $4D_{16}$ c'est à dire 77 en décimal.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1																
2	Espace	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	

Figure 1 Table ASCII

On a reçu le message suivant codé en ASCII :

message = (0x47, 0x65, 0x6F, 0x72, 0x67, 0x65, 0x20, 0x42, 0x6F, 0x6F, 0x6C, 0x65)

- 1) Quel est le type de la variable message ?
- 2) Décodez le message.
- 3) En UTF-8, le codage des caractères coïncide avec l'ASCII pour les 128 premiers caractères. Les autres caractères tels que "é", "€" sont codés par plusieurs octets.

Le message suivant, donné en hexadécimal, a été relevé dans un fichier codé en UTF-8.

message_2 = (0x43, 0x6F, 0x64, 0xC3, 0xA9, 0x20, 0x65, 0x6E, 0x20, 0x55, \n 0x54, 0x46, 0x2D, 0x38)

Il contient uniquement des caractères de la table ASCII à l'exception d'un "é".

- a. Quelle est la séquence d'octets qui représente le "é" ?
- b. Si le message_2 avait été interprété en latin-1 (table ci-après), qu'est-ce qui se serait affiché ?

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	<i>positions inutilisées</i>															
1																
2	Espace	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
8	<i>positions inutilisées</i>															
9																
A	␣	ı	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	®	¯	
B	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Figure 2 Table latin-1

Exercice 3 (9 points)

Un carré d'ordre n est un tableau carré contenant n^2 entiers strictement positifs. On dit qu'un carré d'ordre n est magique si :

il contient tous les nombres entiers $1, 2, 3, 4, 5, \dots, n^2$ depuis 1 jusqu'à n^2 inclus. De plus, il faut que :

- La somme des nombres sur chaque ligne soit la même.
- La somme des nombres sur chaque colonne soit la même.
- La somme des nombres sur chaque diagonale soit la même.
- Les trois sommes précédentes (lignes, colonnes, diagonales) soient les mêmes.

Exemple 1 : le carré contient tous les entiers de 1 à 3^2 inclus et toutes les sommes valent 15.

15	15	15	15
15	2	7	6
15	9	5	1
15	4	3	8

On modélise le carré $n = 3$ par la variable `carre3 = [[2, 7, 6], [9, 5, 1], [4, 3, 8]]`

Exemple 2 : le carré magique suivant est un carré d'ordre $n = 4$

4	5	11	14
15	10	8	1
6	3	13	12
9	16	2	7

Pour le modéliser, on crée la variable

```
carre4 = [[4, 5, 11, 14], [15, 10, 8, 1], [6, 3, 13, 12], [9, 16, 2, 7]]
```

- 1) a. Quel est le type de la variable `carre4` ?
b. Quelle est la valeur de `len(carre4)` ?
c. Quelle est la valeur de `carre3[1]` ?
d. Quelle est la valeur de `carre3[0][2]` ?
e. Quelle expression permet de récupérer la valeur 3 qui est dans la variable `carre4` ?

2) On donne la fonction ci-contre :

- a. Que renvoie `somme_ligne(carre4, 2)` ?
b. A quoi sert la fonction `somme_ligne` ?

3) Proposer, en langage Python, une fonction `verifie_ligne`, prenant en paramètres

- une liste de listes représentant un tableau carré de nombres entiers
- un entier égal au côté du tableau carré

et qui renvoie un booléen indiquant l'égalité des sommes sur toutes les lignes ou non.

Par exemple :

```
>>> verifie_lignes(carre3, 3)
True
>>> verifie_lignes(carre4, 4)
True
```

```
def somme_ligne(carre, i):
    """
    Paramètres :
    -----
    carre : de type liste de listes.
            Elle représente un tableau carré de nombres.
    i : de type entier.
    """
    somme = 0
    for nombre in carre[i]:
        somme = somme + nombre
    return somme
```