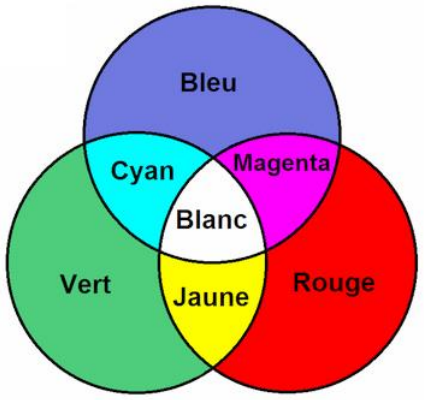
Spécialité NSI Première	**DEVOIR SURVEILLE DE**	Lundi 12 février 2024
Lycée d'Avesnières	**NSI**	Durée : 2 h
Année scolaire 2023-2024	**N° 4**	Calculatrice interdite

L'énoncé complet est à rendre avec la copie.

NOM : ..... Prénom : .....

Exercice 1 (3 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.



Synthèse additive

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 :

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

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 ?

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 ^ .

```
a = 0b1100
b = 0b1010

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

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

Donner la couleur obtenue si on fait les opérations suivantes :

a. bleu | rouge

b. magenta & cyan

c. vert ^ blanc

Exercice 2 (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 dite 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,\
             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	*positions inutilisées*															
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	*positions inutilisées*															
9																
A	NBSP	¡	¢	£	¤	¥	¦	§	¨	©	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 (4,5 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 $1, 2, 3, 4, 5, \ldots, n^2$ 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` ?

```
def somme_ligne(carre, i):

    """

    Paramètres :
    ------------
    carre : de type liste de listes.
            Elle représente un tab-
            leau carré de nombres.

    i : de type entier.

    """

    somme = 0
    for nombre in carre[i]:
        somme = somme + nombre

    return somme
```

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
```

Exercice 4 (7 points)

Partie 1

On dispose d'un fichier csv nommé `capitales.csv` dont voici les 20 première lignes

nom_pays	nom_capitale	latitude	longitude	code_pays	continent
Somaliland	Hargeisa	9.55	44.050000	NULL	Africa
South Georgia and South Sandwich Islands	King Edward Point	-54.283333	-36.500000	GS	Antarctica
French Southern and Antarctic Lands	Port-aux-Français	-49.35	70.216667	TF	Antarctica
Palestine	Jerusalem	31.766666666666666	35.233333	PS	Asia
Aland Islands	Mariehamn	60.116667	19.900000	AX	Europe
Nauru	Yaren	-0.5477	166.920867	NR	Australia
Saint Martin	Marigot	18.0731	-63.082200	MF	North America
Tokelau	Atafu	-9.166667	-171.833333	TK	Australia
Western Sahara	El-Aaiún	27.153611	-13.203333	EH	Africa
Afghanistan	Kabul	34.516666666666666	69.183333	AF	Asia
Albania	Tirana	41.316666666666667	19.816667	AL	Europe
Algeria	Algiers	36.75	3.050000	DZ	Africa
American Samoa	Pago Pago	-14.266666666666667	-170.700000	AS	Australia
Andorra	Andorra la Vella	42.5	1.516667	AD	Europe
Angola	Luanda	-8.833333333333334	13.216667	AO	Africa
Anguilla	The Valley	18.216666666666665	-63.050000	AI	North America
Antigua and Barbuda	Saint John's	17.116666666666667	-61.850000	AG	North America

Argentina	Buenos Aires	-34.583333333333336	-58.666667	AR	South America
Armenia	Yerevan	40.166666666666664	44.500000	AM	Europe
Aruba	Oranjestad	12.516666666666667	-70.033333	AW	North America
Aland Islands	Mariehamn	60.11666	19.90000	AX	Europe

On dispose également d'un fichier Python exercice.py placé dans le même répertoire que le fichier `capitales.csv` :

```
import csv

#ouverture du fichier csv
def lecture_fichier(nom_fichier):
    with open(nom_fichier, mode='r', encoding='utf-8-sig') as fichier_ouvert:
        return [ligne for ligne in csv.reader(fichier_ouvert)]
```

1) Quels sont les descripteurs de la table ?
2) On souhaite récupérer le contenu du fichier csv dans le programme Python .

Quelle instruction Python faut-il écrire afin de lire le fichier capitales.csv et d'affecter son contenu dans la variable appelée `table_capitales` ?

3) Quel est le type de la variable `table_capitales` ?
4) On ajoute la fonction selectionner(table, critere) au programme Python :

```
def selectionner(table, critere):
    selection=[]
    for i in range(1,len(table)):
        if critere(table[i]):
            selection.append(table[i])
    return selection
```

On exécute la ligne :

`recherche_1 = selectionner(table_capitales, lambda x: float(x[2]) < 0)`
Que va contenir la variable `recherche_1` ? Expliquez votre réponse.

5) Grâce à la fonction `selectionner(table, critere)` du programme, donnez le critère de sélection à écrire en argument pour renvoyer uniquement les capitales situées sur le continent sud-américain et affecter le résultat de la recherche dans une variable nommée `recherche_2`.

6) Grâce à la fonction `selectionner(table, critere)` du programme, donnez le critère de sélection à écrire en argument pour renvoyer uniquement les capitales d'Amérique du sud située dans l'hémisphère Nord dans une variable nommée `recherche_3`.

Partie 2

On dispose d'un fichier `csv` nommé `sacs.csv` dans le même répertoire que le fichier `exercice.py`.

On a importé dans le programme Python le fichier `sacs.csv` en exécutant l'instruction

```
table_sacs = lecture_fichier('sacs.csv').
```

Cela a permis d'affecter à la variable `table_sacs` la valeur :

```
[['id', 'nom', 'longueur', 'largeur', 'hauteur', 'couleur', 'poids', 'prix_HT', 'fabrique
en France'], ['1', 'trousse', '20', '6', '7', 'bleu', '150', '18', 'oui'], ['2',
'polochon', '50', '25', '25', 'rouge', '450', '79', 'oui'], ['3', 'cabas', '35', '14',
'30', 'bleu', '330', '95', 'non'], ['4', 'besace', '24', '21', '9', 'marron', '500', '84',
'oui'], ['5', 'banane', '23', '9', '13', 'noir', '260', '28', 'oui'], ['6', 'baluchon',
'61', '38', '22', 'beige', '900', '145', 'non'], ['7', 'sac à dos', '30', '40', '18',
'gris', '375', '50', 'oui'], ['8', 'seau', '24', '25', '13.5', 'noir', '430', '110',
'oui'], ['9', 'wrist bag', '27', '16.5', '3.5', 'bleu', '125', '28', 'non'], ['10',
'minaudiere', '23', '5', '8', 'noir', '160', '35', 'oui']]
```

1) Écrire une instruction Python qui permet d'affecter à une variable nommée `les_descripteurs` la liste des descripteurs de la table `table_sacs`.

2) Il est possible de trier la table grâce à la fonction `sorted()` intégrée à Python et à l'utilisation d'une fonction lambda. Que va contenir la liste de listes s1 après l'instruction suivante ?

```
s1 = sorted(table_sacs[1:], key=lambda ligne: float(ligne[2]))
```

On ne demande pas d'écrire la liste de listes s1 complètement, mais de décrire ce qu'elle contient.

3) Même question avec

```
s2 = sorted(table_sacs[1:], key=lambda ligne: float(ligne[7]), reverse=True)
```

4) Écrire une instruction Python qui permet d'affecter à une variable nommée ma_selection les cinq sacs les plus chers.

5) Soit la variable `s3` la liste de listes `table_sacs` rangée par ordre alphabétique des noms de sacs.

   a. Écrire l'instruction Python permettant d'obtenir `s3`.

   b. Écrire l'instruction Python permettant d'obtenir à partir de la table triée par noms `s3` uniquement les sacs de couleur noire.

Exercice 5 (3 points)

On souhaite concaténer les deux tables ci-contre qui contiennent les données recueillies lors d'une expérience scientifique franco-américaine.

Table 1

id_exp	T	tps
1	20.2	56
2	15.5	85
3	18.6	120

T : °C
tps : s

Table 2

id_exp	T	tps
1	53.6	210
2	62.6	81
3	66.2	70

T : °F
tps : s

T désigne une température et tps une durée en seconde. La difficulté est que les températures T dans les mesures françaises sont en degrés Celsius et que les températures T dans les mesures américaines sont en degrés Fahrenheit. Mais on connait la formule qui permet de transformer une température Fahrenheit T_F en une température Celsius T_C :

$$T_C = (T_F - 32) \times \frac{5}{9}$$

On a créé deux listes de listes nommées table1 et table2 contenant les informations des deux tables décrites ci-dessus.

```
table1 = [['id_exp', 'T', 'tps'], [1, 20.2, 56], [2, 15.5, 85], [3, 18.6, 120]]
table2 = [['id_exp', 'T', 'tps'], [1, 53.6, 210], [2, 62.6, 81], [3, 66.2, 70]]
```

1) Les deux tables ont-elles les mêmes descripteurs ?

2) Betty a écrit la fonction betty(t1, t2) suivante :

```
def betty(t1, t2):
    """
    Réalise la concaténation de t1 et t2.

    Paramètres :
    ------------
            t1 et t2 de type liste de listes.

    Retourne :
    ----------
            t1 + t2[1:] de type liste de listes.

    """

    return t1 + t2[1:]
```

puis elle essaye sa fonction :

```
resultat_betty = betty(table1, table2)
```

Elle obtient la table resultat_betty qui vaut :

```
[['id_exp', 'T', 'tps'], [1, 20.2, 56], [2, 15.5, 85], [3, 18.6, 120], [1, 53.6, 210], [2, 62.6, 81], [3, 66.2, 70]]
```

Est-ce que cette façon de procéder est satisfaisante ? Expliquez.

3) John veut améliorer la fonction de Betty. Pour cela, il a écrit la fonction `fusion(t1, t2)` de la façon suivante :

```
def fusion(t1, t2):
    """
    Réalise la conversion des températures de t2 de degrés F en degrés C.
    Numérote à partir de 4 les id_exp de t2.
    Concatène t1 et t2.

    Paramètres :
    ------------
            t1 et t2 de type liste de listes.

    Retourne :
    ----------
            t1 + t2[1:] de type liste de listes.

    """

    #  Conversion des températures de t2 en coupant sa ligne des descripteurs.
    for ligne in t2[1:]:
        ...

    #  Renumérote les id_exp de t2 en coupant sa ligne des descripteurs.
    for ligne in t2[1:]:
        ...

    return t1 + t2[1:]
```

puis il essaye sa fonction :

```
resultat_john = fusion(table1, table2)
```

Sur votre copie, écrivez **uniquement les deux boucles for** de façon à obtenir `resultat_john` qui vaut :

```
[['id_exp', 'T', 'tps'], [1, 20.2, 56], [2, 15.5, 85], [3, 18.6, 120], [4, 12.0, 210], [5, 17.0, 81], [6, 19.0, 70]]
```

Remarque : Améliorez le résultat de la conversion en degrés Celsius en arrondissant à 1 décimale à l'aide de la fonction intégrée dans Python `round(a, n)` qui arrondit le nombre a à 10^{-n} près.