

# Codage de l'information

mercredi 17 décembre 2025 de 08h00 à 10h00 – durée 2h00

**NUMERO PLACE:** .....

**GROUPE:**

- |                                    |                                    |                                    |                                  |
|------------------------------------|------------------------------------|------------------------------------|----------------------------------|
| <input type="checkbox"/> MI11 L.N. | <input type="checkbox"/> MI14 L.N. | <input type="checkbox"/> MI22 B.D. | <input type="checkbox"/> inconnu |
| <input type="checkbox"/> MI12 H.K. | <input type="checkbox"/> MI15 M.K. | <input type="checkbox"/> MI23 H.K. |                                  |
| <input type="checkbox"/> MI13 C.D. | <input type="checkbox"/> MI21 B.P. | <input type="checkbox"/> MI24 L.W. |                                  |

Nom : \_\_\_\_\_  
 Prénom : \_\_\_\_\_  
 NIP : \_\_\_\_\_

Réservé au correcteur:

Q.1:    Q.2:    Q.3:    Q.4:    Q.5:    Q.6:    Q.7:    Q.8:    Q.9:    Q.10:    Q.11:    Q.12:    Q.13:    Q.14:

Aucun document n'est autorisé. Les calculatrices, téléphones, dictionnaires électroniques sont interdits.

Toute réponse doit être justifiée. Si la résolution de la question nécessite un calcul, faites figurer ce calcul sur la copie.

Répondez directement sur le sujet après avoir complété soigneusement l'entête. Cochez votre groupe de codage. L'oubli sera pénalisé.

Attachez une grande importance au soin, utilisez du brouillon, car il ne sera pas redistribué d'exemplaire du sujet.

Vous pouvez **demandeur un intercalaire** si besoin, mais n'oubliez pas d'y inscrire votre numéro de place. Les brouillons sont refusés.

## Exercice 1 Questions diverses

**Q.1:** Quelle formule mathématique calcule le *logarithme* de  $x$  en base  $b$ , en utilisant le *logarithme naturel*  $\ln$  ainsi qu'une opération arithmétique.

.....  
 .....

**Q.2:** Quelle formule mathématique calcule la taille de l'écriture d'un nombre de valeur  $N$ , lorsque cette écriture est réalisée en base  $b$ ? Une démonstration est demandée.

.....  
 .....  
 .....  
 .....  
 .....  
 .....  
 .....

**Q.3:** Calculez  $(\overline{1001001001})_2 \times (\overline{111})_2$  en base 2.  
 Prouvez que,  $\forall n \in \mathbb{N}$ , si  $2^n - 1$  est premier, alors  $n$  est premier.

.....  
 .....  
 .....  
 .....  
 .....  
 .....  
 .....  
 .....

**Q.4:**

- (a) Calculez l'écriture de l'entier décimal 125 en base 8,
- (b) Sans divisions supplémentaires, donnez son écriture en base 2 à partir de l'écriture en base 8,
- (c) Sans calculs supplémentaires, donnez son écriture en base 16 à partir de l'écriture en base 2,
- (d) Vérifiez que l'écriture en base 16 corresponde bien à la valeur décimale 125.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

**Q.5:** Donnez, en base 2 et sur 8 bits, l'écriture de l'entier décimal relatif -125 en,

- (a) signe + valeur absolue
- (b) complément à deux.

.....

.....

.....

.....

.....

.....

.....

.....

**Q.6:** On cherche une écriture du nombre décimal 25,2

- (a) Donnez sa partie entière en base 2, puis le calcul de sa partie décimale en base 2.
- (b) Donnez son écriture exacte (avec répétition éventuelle soulignée) en virgule fixe, puis flottante.
- (c) Donnez sa représentation hexadécimale IEEE 754 demi-précision, exacte sinon tronquée.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

### Exercice 3 Unicode, UTF-8

Ci dessous, nous vous proposons une capture hexadécimale d'un fichier texte

```
00000000  4c 65 20 63 61 72 61 63 74 c3 a8 72 65 20 42 72 |Le caract..re Br|
00000010  61 69 6c 6c 65 20 64 65 20 6c 61 20 6c 65 74 74 |aille de la lett|
00000020  72 65 20 57 20 65 73 74 20 6e 6f 74 c3 a9 20 70 | [A COMPLETER] |
00000030  61 72 20 e2 a0 ba                                     |ar ...|
00000036
```

**Q.7:** Pour le seul symbole accentué de la 1ère ligne, quel est l'encodage utilisé et la taille de l'encodage ; Calculez ensuite son *point de code* Unicode (hexadécimal).

.....  
.....  
.....  
.....

**Q.8:** Décodez la 3ème ligne, en incluant les codes UTF-8 et leurs *points de code* Unicode respectifs

.....  
.....  
.....  
.....

**Q.9:**

- (a) Quelle est la taille de codage utilisée par les derniers octets de la 4ème ligne ?
- (b) Donnez le principe permettant de décoder le (ou les) symbole(s).
- (c) Calculez le *point de code* Unicode (hexadécimal) du (ou des) symbole(s) associé(s).
- (d) Retrouvez puis vérifiez le(s) symbole(s) associé(s) grâce à l'annexe.

.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....

Désormais, on s'intéresse aux symboles de la *table 1.7.1 en annexe*.

**Q.10:** Proposez une fonction qui, à partir d'un paramètre entier  $u$  compris entre 0 et 63, retourne la valeur de son *point de code* Unicode (il s'agit d'une valeur entière). On demandera de privilégier les opérations logiques aux opérations arithmétiques, en les justifiant.

.....  
.....  
.....  
.....  
.....  
.....  
.....

**Q.11:** Proposez des formules logiques permettant, à partir d'un point de code Unicode  $p$  (résultat de la fonction précédente), de donner la valeur des différents octets UTF-8 associés.

.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....

On souhaite déduire des informations sur les six bosses *présentes ou absentes* dans chacun des symboles Braille de la *table 1.7.1 en annexe*, cela à partir uniquement de leur *point de code* Unicode. Tout d'abord, nous proposons quelques questions d'observation sur les symboles Braille.

**Q.12:**

- (a) à une certaine position, la bosse est alternativement absente puis présente à chaque fois qu'on passe au point de code suivant. Quelle est cette position ? À quel bit du point de code  $p$  associé correspond-il ?
- (b) Mêmes questions pour la bosse dont la présence change tous les deux points de code.
- (c) Mêmes questions pour la bosse qui change *le moins régulièrement* dans tous les symboles.

.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....

**Q.13:** Proposez un prédicat `bosse_presente`, qui, à partir d'un point de code Unicode  $p$  d'un des 64 symboles, et d'un entier  $u$  compris entre 0 et 5, renvoie un booléen qui est vrai si la bosse portant ce numéro est présente. (indication : `bool(0)` vaut `False` et `bool(1)` vaut `True`).

.....  
.....  
.....  
.....  
.....  
.....

**Q.14:** Avec le prédicat précédent, comment savoir si la ligne du haut (bosse 0 et bosse 3) d'un symbole Braille - (a) est complète (deux bosses hautes présentes) - (b) est non vide (au moins une bosse en haut) - (c) dispose d'une et d'une seule bosse en haut ?

.....  
.....  
.....  
.....  
.....  
.....

# 1 Annexe

## 1.1 décimal, binaire, octal, hexadécimal

hexadécimal	0	1	2	3	4	5	6	7
binaire	0000	0001	0010	0011	0100	0101	0110	0111
octal	0	1	2	3	4	5	6	7
décimal	0	1	2	3	4	5	6	7
hexadécimal	8	9	A	B	C	D	E	F
binaire	1000	1001	1010	1011	1100	1101	1110	1111
octal	10	11	12	13	14	15	16	17
décimal	8	9	10	11	12	13	14	15

Notation Python :  $(1AF2)_{16} = 0x1AF2$        $(172)_8 = 0o172$        $(1010)_2 = 0b1010$

## 1.2 code ASCII

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOt	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SP	!	”	#	\$	%	&	'	(	)	*	+	,	-	.	/
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	{		}	~	DEL

## 1.3 codage UTF-8

Octets	Premier	Dernier	Octet 1	Octet 2	Octet 3	Octet 4
1	U+0000	U+7F	0xxxxxxx			
2	U+0080	U+07FF	110xxxxx	10xxxxxx		
3	U+0800	U+FFFF	1110xxxx	10xxxxxx	10xxxxxx	
4	U+10000	U+10FFFF	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx

## 1.4 memento IEEE 754 demi-précision

- taille de la donnée : seize bits
- signe : codé sur un bit (bit 15)
- $E$  : codé sur 5 bits (bits 10 à 14)
- partie fractionnaire de la mantisse des nombres normalisés : codée sur 10 bits (bits 0 à 9)
- biais : 15
- plage des exposants  $e$  des nombres normalisés:  $[-14, 15]$

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
s	$E_4$	$E_3$	$E_2$	$E_1$	$E_0$	$m_{-1}$	$m_{-2}$	$m_{-3}$	$m_{-4}$	$m_{-5}$	$m_{-6}$	$m_{-7}$	$m_{-8}$	$m_{-9}$	$m_{-10}$

## 1.5 puissances

i	$2^i$	$5^i$	$2^{-i} = 0.5^i$
0	1	1	1.0
1	2	5	0.5
2	4	25	0.25
3	8	125	0.125
4	16	625	0.0625
5	32	3125	0.03125
6	64	15625	0.015625
7	128	78125	0.0078125
8	256	390625	0.00390625
9	512	1953125	0.001953125
10	1024	9765625	0.0009765625

## 1.6 opérations logiques et décalages

opération	notation mathématique	exemple	opérateur en python
et bit à bit	$\wedge$	$25 \wedge 13$ vaut 9	<code>&amp;</code>
ou bit à bit	$\vee$	$25 \vee 13$ vaut 29	<code> </code>
ou exclusif bit à bit	$\oplus$	$25 \oplus 13$ vaut 20	<code>^</code>
décalage à droite	$\gg$	$256 \gg 5$ vaut 8	<code>&gt;&gt;</code>
décalage à gauche	$\ll$	$1 \ll 5$ vaut 32	<code>&lt;&lt;</code>

## 1.7 Braille 6

### 1.7.1 points de code [0x2800..0x283F]

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x280	⠠	⠡	⠢	⠣	⠤	⠥	⠦	⠧	⠨	⠩	⠪	⠫	⠬	⠭	⠮	⠯
0x281	⠰	⠱	⠲	⠳	⠴	⠵	⠶	⠷	⠸	⠹	⠺	⠻	⠼	⠽	⠾	⠿
0x282	⠠	⠡	⠢	⠣	⠤	⠥	⠦	⠧	⠨	⠩	⠪	⠫	⠬	⠭	⠮	⠯
0x283	⠰	⠱	⠲	⠳	⠴	⠵	⠶	⠷	⠸	⠹	⠺	⠻	⠼	⠽	⠾	⠿

### 1.7.2 numéros et positions des bosses

	gauche	droite
haut	bosse 0 - en haut à gauche	bosse 3 - en haut à droite
milieu	bosse 1 - au milieu à gauche	bosse 4 - au milieu à droite
bas	bosse 2 - en bas à gauche	bosse 5 - en bas à droite

### 1.7.3 Braille en français

a	b	c	d	e	f	g	h	i	j	Utilisation des quatre bosses supérieures
k	l	m	n	o	p	q	r	s	t	Ajout de la bosse 2 (en bas à gauche) aux lettres de la première ligne
u	v	x	y	z	ç	é	à	è	ù	Ajout des bosses 2 et 5 (en bas à droite) aux lettres de la première ligne
â	ê	î	ô	û	ë	ï	ü	œ	w	Ajout de la bosse 5 aux lettres de la première ligne