

**Exercice 1 :** Décoder la suite d'octets suivants, qui représente un texte encodé en ASCII.

(Rappel : 'A' correspond au point de code décimal 65 et 'a' correspond au point de code décimal 97)

01100101 01101110 01100011 01101111 01100100 01100001 01100111 01100101

**Exercice 2 :** Voici les premiers octets d'un petit fichier de taille 1427 octets, lus avec l'éditeur de fichier binaire xxd

>> 89 50 4E 47 0d 0a 1a 0a 00 00 00 0d 49 48 44 52 00 00 00 32 00 00 00 32 08 06 00 00 00 1e 3f 88

En décodant les premiers octets, avec l'encodage ASCII, deviner de quel type de fichier il s'agit.

**Exercice 3 :** Jean a envoyé un message à Pierre mais, à la réception, le message est un peu étrange ...

1°) Expliquer ce qui s'est passé.

2°) Dans la suite d'octets transmise, repérer les octets qui sont mal décodés et expliquer comment les décoder.

texte saisi par Jean >> j'ai été reçu à l'examen

Suite d'octets envoyée dans le réseau >> 01101010 00100111 01100001 01101001 00100000 11000011 10101001 01110100 11000011 10101001 00100000 01110010 01100101 11000011 10100111 01110101 00100000 11000011 10100000 00100000 01101100 00100111 01100101 01111000 01100001 01101101 01100101 01101110

texte reçu par Pierre >> j'ai Ã©tÃ© reÃ§u Ã l'examen

**Exercice 4 :** Donner l'encodage en UTF-8 des caractères ci-dessous à partir de leur point de code unicode.

caractère	point de code	encodage UTF-8
a	U+0061	
É	U+00C9	
î	U+00EE	
ü		C3 BC

caractère	point de code	encodage UTF-8
☐	U+2680	
♟	U+2658	
\n	U+0A	
Å		E2 84 AB

**Exercice 5 :** On dispose de 2 fichiers enregistrés contenant le même texte mais avec un encodage différent : l'un en UTF-8 et l'autre en ISO-8859-15. **Voici le texte →**

**Ça va ?  
Vœux**

☒ **Comparer** la taille en octets de chaque fichier à la longueur du texte ('œ' est un unique caractère)

	contenu du fichier en hexadécimal	encodage
Fichier 1	c3 87 61 20 76 61 20 3f 0a 56 c5 93 75 78	
Fichier 2	c7 61 20 76 61 20 3f 0a 56 bd 75 78	

☒ **Associer** à chaque caractère son encodage et **retrouver** le point de code Unicode de chaque caractère.

	Ç	a		v	a		?		V	œ	u	x
Encodage ISO-8859-15												
Encodage UTF-8												
Point de code Unicode												

**Exercice 6 :** Décoder la suite d'octets suivants, qui représente un texte encodé en UTF8.

(Pour les caractères codés sur plus d'un octet, utiliser la fonction Python chr(x) où x est un nombre entier)

01010011 01110000 11000011 10101001 00100000 01001110 01010011 01001001 00100000 11100010 10011001 10100101

## CORRECTION

**Exercice 1 :** Décoder la suite d'octets suivants, qui représente un texte encodé en ASCII.

(Rappel : 'A' correspond au point de code décimal 65 et 'a' correspond au point de code décimal 91)

01100101 01101110 01100011 01101111 01100100 01100001 01100111 01100101  
e n c o d a g e

**Exercice 2 :** Voici les premiers octets d'un petit fichier de taille 1427 octets, lus avec l'éditeur de fichier binaire xxd

>> 89 50 4E 47 0d 0a 1a 0a 00 00 00 0d 49 48 44 52 00 00 00 32 00 00 00 32 08 06 00 00 00 1e 3f 88

En décodant les premiers octets, avec l'encodage ASCII, deviner de quel type de fichier il s'agit.

octet : 89 (hexa)	137 (décimal)	car :
octet : 50 (hexa)	80 (décimal)	car : P
octet : 4e (hexa)	78 (décimal)	car : N
octet : 47 (hexa)	71 (décimal)	car : G
octet : 0d (hexa)	13 (décimal)	car :
octet : 0a (hexa)	10 (décimal)	car :
octet : 1a (hexa)	26 (décimal)	car :
octet : 0a (hexa)	10 (décimal)	car :

Un fichier PNG commence par une signature de 8 octets représenté par les valeurs décimales suivantes :

137 80 78 71 13 10 26 10

ou en hexadécimal :

89 50 4E 47 0D 0A 1A 0A.

La suite du fichier est décomposée en plusieurs parties de longueurs variables, appelées chunk.

```
ch='89 50 4e 47 0d 0a 1a 0a 00 00 00 0d 49 48 44 52'.split(' ')
for el in ch:
    o1=int(el[:2],16)
    print(f"octet : {el} (hexa)\t{o1} (décimal)\tcar : {chr(o1)}")
```

**Exercice 3 :**

texte saisi par Jean >> j'ai été reçu à l'examen

Suite d'octets envoyée dans le réseau >> 01101010 00100111 01100001 01101001 00100000 11000011 10101001  
01110100 11000011 10101001 00100000 01110010 01100101 11000011 10100111 01110101 00100000 11000011  
10100000 00100000 01101100 00100111 01100101 01111000 01100001 01101101 01100101 01101110

En hexadecimal >> 6a 27 61 69 20 c3 a9 74 c3 a9 20 72 65 c3 a7 75 20 c3 a0 20 6c 27 65 78 61 6d 65 6e

texte reçu par Pierre >> j'ai Ã©tÃ© reÃ§u Ã l'examen

Le texte a été encodé en UTF8 et décodé avec un autre encodage obsolète : ISO-8859-1 (latin1) ou ISO-8859-15 (latin9) ou le plus probablement cp1252 (abusivement appelé ANSI)

Les octets mal décodés sont ceux qui ne correspondent pas à un caractère dont le point de code est inférieur à 127.

On les repère parce qu'ils commencent par 1 ou 11 au lieu de 0.

Ils sont codés sur deux octets en utf8 et ces deux octets sont alors décodés comme deux caractères de la table cp1252.

```
#-----
# Problème d'encodage : un exemple
#-----

# Jean écrit un mail
texte = "j'ai été reçu à l'examen"
print('texte saisi par Jean :')
print(texte, '\n')
# son OS l'encode pour le faire passer sur le réseau
octets_transmis = texte.encode(encoding="utf-8")

print("Suite d'octets envoyée dans le réseau : ")
for octet in octets_transmis:
    print(f"{octet:08b}", end=" ")
print('\n')
for octet in octets_transmis:
    print(f"{octet:02x}", end=" ")
print('\n')

# Pierre reçoit le binaire mais son ordi est un vieux Windows mal configuré
texte_recu = octets_transmis.decode(encoding="cp1252")

# Pierre voit ceci
print('texte reçu par Pierre : ')
print(texte_recu)
```

**Exercice 4 : Donner** l'encodage en UTF-8 des caractères ci-dessous à partir de leur point de code unicode.

car	point de code	encodage UTF-8
a	<b>U+0061</b> 01100001	<b>61</b> 01100001
É	<b>U+00C9</b> 11001001	<b>C3 89</b> 11000011 10001001
î	<b>U+00EE</b> 11101110	<b>C3 AE</b> 11000011 10101110
ü	<b>U+00FC</b> 11111100	<b>C3 BC</b> 11000011 10111100

car	point de code	encodage UTF-8
☐	<b>U+2680</b> 10011010000000	<b>E2 9A 80</b> 11100010 10011010 10000000
♙	<b>U+2658</b> 10011001011000	<b>E2 99 98</b> 11100010 10011001 10011000
\n	<b>U+000A</b> 00001010	<b>0A</b> 00001010
Å	<b>U+0021B</b> 10000100101011	<b>E2 84 AB</b> 11100010 10000100 10101011

**Exercice 5 :** On dispose de 2 fichiers enregistrés contenant le même texte mais avec un encodage différent : l'un en UTF-8 et l'autre en ISO-8859-15. **Voici le texte →**

**Ça va ?  
Vœux**

☒ **Comparer** la taille en octets de chaque fichier à la longueur du texte ('œ' est un unique caractère)

	contenu du fichier en hexadécimal	encodage
<b>Fichier 1</b>	c3 87 61 20 76 61 20 3f 0a 56 c5 93 75 78	<b>UTF-8</b>
<b>Fichier 2</b>	c7 61 20 76 61 20 3f 0a 56 bd 75 78	<b>ISO-8859-15</b>

☒ **Associer** à chaque caractère son encodage et **retrouver** le point de code Unicode de chaque caractère.

	Ç	a		v	a		?	\n	V	œ	u	x
Encodage ISO-8859-15	c7	61	20	76	61	20	3f	0a	56	bd	75	78
Encodage UTF-8	c3 87	61	20	76	61	20	3f	0a	56	c5 93	75	78
Point de code Unicode (décimal)	199	97	32	118	97	32	63	10	86	339	117	120

```
# pour afficher la suite d'octets en hexadécimal, séparés par un espace
ch = 'Ça va ? Vœux'
octet_str = ch.encode('utf8').hex()

ch_hex = ''
i = 1
for octet in octet_str:
    ch_hex += octet
    if i % 2 == 0 : ch_hex += ' '
    i += 1
print(ch_hex)
```

**Exercice 6 :** Décoder la suite d'octets suivants, qui représente un texte encodé en UTF8 >> **Spé NSI ♥**

01010011 01110000 11000011 10101001 00100000 01001110 01010011 01001001 00100000 11100010 10011001 10100101

**Mauvais décodage avec latin9**

01010011 01110000 11000011 10101001 00100000 01001110 01010011 01001001 00100000 11100010 10011001 10100101  
83 112 195 169 32 78 83 73 32 226 153 165  
S p Ã Ø N S I â ¥

**Bon décodage avec utf8**

01010011 01110000 11000011 10101001 00100000 01001110 01010011 01001001 00100000 11100010 10011001 10100101  
83 112 233 32 78 83 73 32 9829  
S p é N S I ♥

```
texte = "Spé NSI ♥"
octets = texte.encode('utf-8')
suite_en_decimal = list(octets)
suite_en_binaire = [bin(octet)[2:].rjust(8, '0') for octet in octets]

print(suite_en_decimal)
print(suite_en_binaire)
print(' '.join(suite_en_binaire))
for n in suite_en_decimal:
    print(f"{n:^9}", end=" ")
print()
for n in suite_en_decimal:
    print(f"{chr(n):^9}", end=" ")
```