

Exercices encodage correction

Christophe Viroulaud

Première - NSI

DonRep 15

1. Exercice 1

2. Exercice 2

3. Exercice 3

4. Exercice 4

Exercice 1

Exercice 2

Exercice 3

Exercice 4

1. Les caractères sont notés en hexadécimal.
2. Attention le 20 est en hexadécimal ; il s'agit donc en binaire de 0010 0000.
3. En décimal $00100000 = 2^5 = 32$
4. VIVE LES VACANCES

1. Exercice 1

2. Exercice 2

3. Exercice 3

4. Exercice 4

Exercice 1

Exercice 2

Exercice 3

Exercice 4

1. U+0040 en binaire : 0100 0000
2. Il suffit d'un octet pour encoder ce caractère en UTF-8.
3. Le point de code de la lettre Ê est U+00CA. En binaire on a : 1100 1010. Il faut alors 2 octets pour encoder ce caractère en UTF-8.

On utilise la suite d'octets : 110xxxxx 10xxxxxx et on remplace les x par les chiffres binaires. On complète avec des 0 :

11000011 10001010

1. Exercice 1

2. Exercice 2

3. Exercice 3

4. Exercice 4

Exercice 1

Exercice 2

Exercice 3

Exercice 4

Exercice 3

Pour convertir en binaire il faut effectuer des divisions successives par 2.

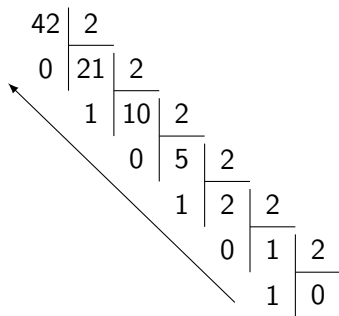


FIGURE 1 - $42_{10} = 101010_2$

Exercice 1

Exercice 2

Exercice 3

Exercice 4

```
1 def deci_bin(entier: int) -> str:
2     res = ""
3     while entier > 0:
4         res = str(entier % 2)+res
5         entier = entier//2
6     return res
```


Exercice 1

Exercice 2

Exercice 3

Exercice 4

S	A	L	U	T
83	65	76	85	84

```
1 def decoder(code_car: list) -> str:
2     res = ""
3     for code in code_car:
4         res = res+chr(code)
5     return res
```

```
1 print(decoder([83, 65, 76, 85, 84]))
2 # affiche 'SALUT'
```

Code 1 – Appel de la fonction

1. Exercice 1

2. Exercice 2

3. Exercice 3

4. Exercice 4

Exercice 1

Exercice 2

Exercice 3

Exercice 4

Documentation

```
def ord(c : Text) -> int
```

Return the Unicode code point for a one-character string.

Documentation

```
def hex(i : int) -> str
```

Return the hexadecimal representation of an integer.

```
hex(12648430)
```

```
'0xc0ffee'
```

```
1 def utf8(car: str) -> str:  
2     return hex(ord(car))
```

```
1 >>> utf8("é")  
2 0xE9
```

Code 2 – Appel de la fonction

```
1 def encoder_hexa(phrase: str) -> list:  
2     codes = []  
3     for lettre in phrase:  
4         codes.append(utf8(lettre))  
5     return codes
```

```
1 >>> encoder_hexa("éléphant")  
2 ['0xe9', '0x6c', '0xe9', '0x70', '0x68', '0'  
   x61', '0x6e', '0x74']
```

Code 3 – Appel de la fonction

```
1 def encoder_hexa2(phrase: str) -> list:
2     # crée un tableau à la bonne dimension
3     codes = ["" for _ in range(len(phrase))]
4
5     for i in range(len(phrase)):
6         codes[i] = utf8(phrase[i])
7     return codes
```

Code 4 – Seconde version

```
1 >>> encoder_hexa2("éléphant")
2 ['0xe9', '0x6c', '0xe9', '0x70', '0x68', '0x61', '0x6e', '0x74']
```

Code 5 – Appel de la fonction