Exercices encodage correction

Christophe Viroulaud

Première - NSI

DonRep 15

Sommaire

Exercice 1

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

Sommaire

Exercice 2

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 :

   $$110\mathbf{00011}\ 10\mathbf{001010}$$

Le point de code de la lettre € est U+20AC. En binaire on a : 0010 0000 1010 1100. Il faut alors 3 octets pour encoder ce caractère en UTF-8.
On utilise la suite d'octets : 1110xxxx 10xxxxxx 10xxxxxx et on remplace les x par les chiffres binaires. On complète avec des 0 :

$$1110\mathbf{0010}\ 10\mathbf{000010}\ 10\mathbf{101100}$$

Sommaire

Exercice 3

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

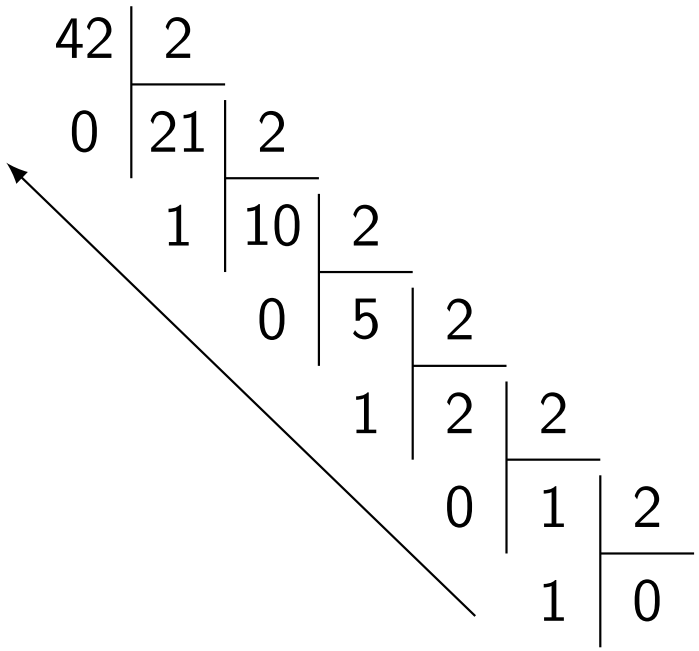

FIGURE 1 – $42_{10} = 101010_2$

```
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
```

S	A	L	U	T
83	65	76	85	84

```python
def decoder(code_car: list) -> str:
    res = ""
    for code in code_car:
        res = res+chr(code)
    return res
```

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

Code 1 – Appel de la fonction

Sommaire

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

```python
def encoder_hexa2(phrase: str) -> list:
    # crée un tableau à la bonne dimension
    codes = ["" for _ in range(len(phrase))]

    for i in range(len(phrase)):
        codes[i] = utf8(phrase[i])
    return codes
```

Code 4 – Seconde version

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

Code 5 – Appel de la fonction