

2026 sujet 22

Question 1

```
def bin2dec(t):
    s = 0
    n = len(t)
    for i in range(n):
        s = s + t[i] * 2**(n-1-i)
    return s
```

Ensuite, il faut calculer la valeur de chaque tuple :

```
>>> bin2dec((0, 1, 0, 0, 1, 1, 0, 1))
77
>>> bin2dec((0, 0, 1, 0, 1, 1, 1, 0))
46
>>> bin2dec((0, 1, 0, 0, 1, 0, 0, 0))
72
>>> bin2dec((0, 1, 1, 0, 0, 0, 0, 1))
97
>>> bin2dec((0, 1, 1, 1, 0, 0, 1, 0))
114
>>> bin2dec((0, 1, 1, 0, 0, 0, 0, 1))
97
```

Puis en regardant la correspondance dans le tableau : M.Hara.

Question 2

```
def qrcode2dec(qrcode):
    liste = []
    for t in qrcode:
        liste.append(bin2dec(t))
    return liste
```

Et le test :

```
assert qrcode2dec(qrcode_fig1) == [77, 46, 72, 97, 114, 97]
```

Question 3

Il y a un problème avec une clé qui n'est pas dans la table ASCII :

```
>>> test_dec2str()
Test 1 réussi!
Test 2 réussi!
```

```

Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "qrcode.py", line 46, in test_dec2str
    print(dec2str(test))
  File "qrcode.py", line 38, in dec2str
    chaine += table_ascii[entier]
KeyError: 233

```

On vérifie donc que le nombre est dans le dictionnaire avant de le convertir :

```

def dec2str(liste_dec):
    """ entrée: liste d'entiers décimaux
        sortie: chaîne de caractère formée des caractères correspondant
        de la table ascii """
    table_ascii = ascii.dict_ascii
    chaine = ""
    for entier in liste_dec:
        if entier in table_ascii :
            chaine += table_ascii[entier]
    return chaine

```

Et cela fonctionne :

```

>>> test_dec2str()
Test 1 réussi!
Test 2 réussi!
Test 3 réussi!

```

Question 4

On essaie :

```

>>> str2qrcode("M.Hara")
[(1, 0, 0, 1, 1, 0, 1), (1, 0, 1, 1, 1, 0), (1, 0, 0, 1, 0, 0, 0),
(1, 1, 0, 0, 0, 0, 1), (1, 1, 1, 0, 0, 1, 0), (1, 1, 0, 0, 0, 0, 1)]

```

Il manque le 0 au début de chaque octet. On peut alors utiliser la méthode `zfill` qui ajoute des zéros à une chaîne de caractères :

```

def str2qrcode(message):
    """
    Convertit une chaîne de caractères en liste de tuples binaires.
    """
    qrcode = []
    table_inverse = {valeur: cle for cle, valeur in ascii.dict_ascii.items()}

    for caractere in message:
        entier = table_inverse.get(caractere, 63)
        binaire_str = bin(entier)[2:].zfill(8)
        ligne = tuple(int(bit) for bit in binaire_str)
        qrcode.append(ligne)

    return qrcode

```

et la sortie est alors correcte :

```
>>> str2qrcode("M.Hara")  
[(0, 1, 0, 0, 1, 1, 0, 1), (0, 0, 1, 0, 1, 1, 1, 0), (0, 1, 0, 0, 1, 0, 0, 0),  
(0, 1, 1, 0, 0, 0, 0, 1), (0, 1, 1, 1, 0, 0, 1, 0), (0, 1, 1, 0, 0, 0, 0, 1)]
```