

2026 sujet 1

Question 1

Non, par exemple l'image [1,2,3] s'écriera [1, 1, 1, 2, 1, 3] en codage RLE. On peut donc avoir des codages plus longs en RLE.

Question 2

```
def decodage_rle(liste_rle):  
    '''Renvoie la liste d'octets obtenue à partir de la liste liste_rle obtenue  
    par compression RLE'''  
    liste = []  
    i = 0  
    while i < len(liste_rle):  
        compte = liste_rle[i]  
        valeur = liste_rle[i+1]  
        for _ in range(compte):  
            liste.append(valeur)  
        i = i + 2  
    return liste
```

Il n'y pas trop d'intérêt à ajouter de nouveaux tests.

Question 3

On exécute les deux commandes suivantes :

```
encoder_decoder_image("bac_nsi_32.png")  
encoder_decoder_image("bac_nsi_256.png")
```

L'image de 256 pixels est déformée, elle est aplatie et allongée.

Question 4

Le problème vient de la fonction enregistrer_octets qui tronque les valeurs supérieures à 255 pour pouvoir les enregistrer dans un octet. Il faut donc modifier codage_rle pour limiter le compte à 255. On ajoute donc la condition $k < 255$ dans le while :

```
def codage_rle2(liste_octets):  
    '''Renvoie une liste d'octets obtenue par compression RLE'''  
    liste_rle = []  
    i = 0  
    while i < len(liste_octets):  
        c = liste_octets[i]  
        k = 1  
        while i+k < len(liste_octets) and liste_octets[i+k] == c and k < 255:  
            k += 1  
        liste_rle.append(k)  
        liste_rle.append(c)  
        i += k  
    return liste_rle
```