de PIL importer Image
de encodage import tout

class Encode:

methode __init__(path, message)

- attribut qui conserve le chemin de l'image
- attribut qui conserve le message
- attributs qui conserve les attributs de l'image (dimension, pixels…)

methode edit_last_bit(number, bit)

- Remplace le dernier bit de <number> par <bit> et renvoie <number> modifié en int

méthode save_image()

- sauvegarde l'image

methode edit_pixels(base2List, x = 0, y = 0)

- Ecris chaque bit de <base2List> sur les pixels de l'image
- Commence à écrire sur le pixel[x,y] définit dans les paramètres de la fonction et se décale vers la droite jusqu'à la fin de la ligne
- Cette méthode utilise la méthode edit_last_bit

methode insert_number_of_caracter()

- Insère la longueur du message sur les 8 premiers pixels de l'image
- Cela permet de savoir la longueur du message quand on voudra lire le texte écrit sur cette image
- cette méthode utilise la méthode edit_pixels pour cela

methode insert_text()

- insère le message dans le texte
- appelle la méthode edit_pixels avec x = 8

class Decode:

methode __init__(path, message)

- attribut que conserve le chemin de l'image
- attribut qui conserve les attributs de l'image (dimension, pixels…)

méthode read_last_bit(number)

Lis et renvoie le dernier bit de <number>

méthode read_pixels(lenText, x = 0, y = 0)

- Lis le dernier bit que chaque pixel et insère ce bit dans une liste qui sera a la fin renvoyée, utilise la méthode read_last_bit

méthode read_number_of_caracter()

- Lis le nombre de caractère du message et le renvoie (int)
- utlise la méthode read_pixels avec lenText = 24

méthode read_text()

- Lis les caractères du message et renvoie la liste de ces caractères (en binaire)

```
fonction main(path, message <- False, key <- False)
    si message différent de False
        si key différent de False
            message <- XOR(text_to_binary(message), key)
        fin si

        encoding <- Encode(message <- message, path <- path)
        encoding.insert_number_of_caracter()
        encoding.insert_text()
        encoding.save_image()

        renvoyer “Le message a bien été encodé dans l’image”

    sinon
        decoding <- Decode(path <- path)
        longueur <- decoding.read_number_of_caracter()
        text <- decoding.read_text(longueur)

        si key différent de False
            text <- XOR(test, key)
        sinon
            text <- binary_to_text(text)
        fin si
        renvoyer text
    fin si
```