Mini projet: stéganographie:

La stéganographie est l'art de la dissimulation: son objet est de faire passer inaperçu un message dans un autre message. Ici par exemple, nous avons caché le message *'Ceci est un message caché dans une image'* dans une image de Maryline:

Le but de ce projet est de cacher du texte dans une image à l'aide d'un fonction Python puis, d'être capable de récupérer le texte caché avec la méthode précédente à l'aide d'une autre fonction Python.

On propose pour cacher du texte dans une image la méthode suivante:

1. Choisir un texte à dissimuler dans une image,
2. Obtenir le code binaire de ce texte en utf-8,
3. Choisir une image en niveaux de gris,
4. Accéder aux valeurs des pixels de l'image (ces dernières sont comprises entre 0 et 255) et les modifier ainsi:
   - 4.1 si la valeur est impaire, la remplacer par l'entier pair précédent (on remarquera qu'ainsi, on n'a plus de pixel ayant pour valeur 255 et que toutes les valeurs en binaire des pixels finissent par 0),
   - 4.2 ajouter à chaque pixel la valeur du bit correspondant au code binaire du texte à dissimuler,
   - 4.3 lorsque tout le texte à dissimuler a été inséré dans l'image, imposer la valeur di pixel à 255 (ceci permettra de stopper le décodage).
5. Sauvegarder alors l'image.

Un peu d'aide sur ce projet:

- des fonctions qui permettent d'encoder du texte en utf-8 et inversement:

```python
import binascii

def text_to_bits(text, encoding='utf-8', errors='surrogatepass'):
    bits = bin(int(binascii.hexlify(text.encode(encoding, errors)), 16))[2:]
    return bits.zfill(8 * ((len(bits) + 7) // 8))

def text_from_bits(bits, encoding='utf-8', errors='surrogatepass'):
    n = int(bits, 2)
    return int2bytes(n).decode(encoding, errors)

def int2bytes(i):
    hex_string = '%x' % i
    n = len(hex_string)
    return binascii.unhexlify(hex_string.zfill(n + (n & 1)))

print(text_to_bits('hello'))
print(text_from_bits('110100001100101011011000110110001101111'))
```

steganographie.py

- le début du code:

```python
from PIL import Image
#creation de la premiere image:
image1 = Image.open("marylin_stegano.jpeg")
(L,H) = image1.size



#creation de la seconde image:
image2 = Image.new("L",(L,H))

secret = 'Ceci est un message cache dans une image'
secret_bin = text_to_bits(secret)
print(secret_bin)
secret_bin = secret_bin + 's'      #du type '01100010111010s' s pour stop



for x in range(L):
    for y in range(H):
        p = image1.getpixel((x,y))
        p = (p//2) * 2
        #on n'obtient que des pixels pairs donc qui finissent par 0 en binaire
        #et aucun pixel ayant pour valeur 255
```

steganographie.py

On réalisera les fonctions code() et decode() permettant de cacher du texte dans une image et de le récupérer. On répondra aux questions suivantes:

1. Quel est le type de secret_bin?
2. A quoi correspond l'opération de la ligne 20: $p = (p//2) * 2$
3. Notre méthode est elle indétectable? Proposer alors une solution permettant de la rendre encore plus discrète.