

La partie choisie est : **Inversion négative-positive (valeur de pixel inversée)** :

```
from PIL import Image, ImageDraw
import numpy as np

# On charge l'image et on la transforme en tableau contenant les couleurs
image_entrée = Image.open("Info/Lenna.png")
image = np.asarray(image_entrée)

# Partie à compléter
image_sortie = 255 - image

# On sauvegarde les images pour pouvoir les afficher
Image.fromarray(image).save("image_entree.png")
Image.fromarray(image_sortie).save("image_sortie.png")
```

---

```
from PIL import Image
import numpy as np

# Charger l'image et la transformer en tableau contenant les couleurs
with Image.open("example.jpg") as img:
    arr = np.array(img)

# Inverser les couleurs de chaque pixel
arr_inv = 255 - arr

# Convertir le tableau numpy en image PIL et enregistrer l'image de sortie
img_inv = Image.fromarray(arr_inv)
img_inv.save("image_sortie.png")
```

```
from PIL import Image
import numpy as np

# Charger l'image et la transformer en tableau contenant les couleurs
img = Image.open("example.jpg")
arr = np.array(img)

# Inverser les couleurs de chaque pixel
arr_inv = 255 - arr

# Créer une nouvelle image à partir du tableau inversé
img_inv = Image.fromarray(arr_inv)

# Enregistrer l'image inversée
img_inv.save("image_sortie.png")
```

---

```
from PIL import Image
import numpy as np

# Ouvrir l'image
img = Image.open("example.jpg")

# Convertir l'image en tableau numpy
img_array = np.array(img)

# Inverser les couleurs de chaque pixel
inv_array = 255 - img_array

# Convertir le tableau numpy en image
inv_img = Image.fromarray(inv_array)
```

```
# Enregistrer l'image inversée
```

```
inv_img.save("output.jpg")
```

---

```
import cv2
```

```
# Ouvrir l'image
```

```
img = cv2.imread("example.jpg")
```

```
# Inverser les couleurs de chaque pixel
```

```
inv_img = cv2.bitwise_not(img)
```

```
# Enregistrer l'image inversée
```

```
cv2.imwrite("output.jpg", inv_img)
```

---

```
import cv2
```

```
# Charger l'image
```

```
img = cv2.imread("example.jpg")
```

```
# Obtenir les dimensions de l'image
```

```
height, width, channels = img.shape
```

```
# Inverser les couleurs de chaque pixel
```

```
for y in range(height):
```

```
    for x in range(width):
```

```
        for c in range(channels):
```

```
            img[y, x, c] = 255 - img[y, x, c]
```

```
# Enregistrer l'image inversée
```

```
cv2.imwrite("output.jpg", img)
```

---

```
from PIL import Image, ImageDraw
```

```
import numpy as np

# On charge l'image et on la transforme en tableau contenant les couleurs
image_entrée = Image.open("Info/Lenna.png")
image = np.asarray(image_entrée)
nb_lignes, nb_colonnes, nb_canaux = image.shape

# Partie à compléter
image_sortie = np.copy(image)
for ligne in range(nb_lignes):
    for col in range(nb_colonnes):
        for canal in range(nb_canaux):
            image_sortie[ligne, col, canal] = 255 - image[ligne, col, canal]

# On sauvegarde les images pour pouvoir les afficher
Image.fromarray(image).save("image_entree.png")
Image.fromarray(image_sortie).save("image_sortie.png")
```

---