Rogner:

```
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,_ = image.shape
# On crée notre image de sortie sous forme de tableau numpy (ici on fait juste une copie de l'image originale)
image_sortie = np.copy(image[163:180, 154:232])
# On sauvegarde les images pour pouvoir les afficher
Image.fromarray(image).save("Info/image_entree.png")
Image.fromarray(image_sortie).save("Info/image_sortie.png")
```

Retourner:

```
print("------Retourner------")
    # 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,_ = image.shape
    # Partie à compléter
    image_sortie = np.copy(image)
    for ligne in range(nb_lignes):
        for col in range(nb_colonnes):
            image_sortie[ligne, col] = image[nb_lignes - 1 - ligne, nb_colonnes - 1 - col]
    # On sauvegarde les images pour pouvoir les afficher
    Image.fromarray(image).save("Info/image_entree.png")
    Image.fromarray(image_sortie).save("Info/image_sortie.png")
```

Inversion de couleur:

```
print("------Inversion de couleur------")
    # 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,_ = image.shape
    # Partie à compléter
    image_sortie = np.copy(image)
    for ligne in range(nb_lignes):
        for col in range(nb_colonnes):
            for i in range(3):
                image_sortie[ligne, col, i] = 255 - image[ligne, col, i]
    # On sauvegarde les images pour pouvoir les afficher
    Image.fromarray(image).save("Info/image_entree.png")
    Image.fromarray(image_sortie).save("Info/image_sortie.png")
```

Ne garder qu'une couleur:

```
print("------Unicolore------")
    color = input("Quelle couleur voulez vous garder (r/g/b):")
    if color == "r":
        r = 1
        g = 0
        b = 0
    if color == "g":
        r = 0
        g = 1
        b = 0
    if color == "b":
        r = 0
        g = 0
        b = 1
    # 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,_ = image.shape
    # Partie à compléter
    image_sortie = np.copy(image)
    for ligne in range(nb_lignes):
        for col in range(nb_colonnes):
            image_sortie[ligne, col] = image[ligne, col] * (r, g, b)
    # On sauvegarde les images pour pouvoir les afficher
    Image.fromarray(image).save("Info/image_entree.png")
    Image.fromarray(image_sortie).save("Info/image_sortie.png")
```

Nuance:

```
print("------Nuance------")
    # 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,_ = image.shape
    # Partie à compléter
    image_sortie = np.copy(image)
    for ligne in range(nb_lignes):
        for col in range(nb_colonnes):
            r = image_sortie[ligne, col] * (1, 0, 0)
            g = image_sortie[ligne, col] * (0, 1, 0)
            b = image_sortie[ligne, col] * (0, 0, 1)
            luminance = (0.2126 * r[0] + 0.7152 * g[1] + 0.0722 * b[2])
            image_sortie[ligne,col] = (luminance, luminance, luminance)
    # On sauvegarde les images pour pouvoir les afficher
    Image.fromarray(image).save("Info/image_entree.png")
    Image.fromarray(image_sortie).save("Info/image_sortie.png")
```

Luminosité:

```
print("------Luminosité------")
    # 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,_ = image.shape
    # Partie à compléter
    image_sortie = np.copy(image)
    for ligne in range(nb_lignes):
        for col in range(nb_colonnes):
            for i in range(3):
                image_sortie[ligne,col,i] =  np.clip(image_sortie[ligne, col, i] + 80, 0, 255)
    # On sauvegarde les images pour pouvoir les afficher
    Image.fromarray(image).save("Info/image_entree.png")
    Image.fromarray(image_sortie).save("Info/image_sortie.png")
```

Fond vert:

```
print("------Fond vert------")
    # On charge l'image et on la transforme en tableau contenant les couleurs
    image_lenna = np.asarray(Image.open("Info/Lenna.png"))
    image_fond_vert = np.asarray(Image.open("Info/fond_vert.png"))
    nb_lignes,nb_colonnes,_ = image_fond_vert.shape
    # Partie à compléter
    image_sortie = np.copy(image_fond_vert)
    for ligne in range(nb_lignes):
        for col in range(nb_colonnes):
            if list(image_fond_vert[ligne,col]) == [0,255,0, 255]:
                for i in range(3):
                    image_sortie[ligne,col, i] = image_lenna[ligne,col, i]
    # On sauvegarde les images pour pouvoir les afficher
    Image.fromarray(image_lenna).save("Info/image_entree.png")
    Image.fromarray(image_sortie).save("Info/image_sortie.png")
```

Remplacer une couleur:

```
print("------Remplacer une couleur------")
    # On charge l'image et on la transforme en tableau contenant les couleurs
    image_entrée = Image.open("Info/fond_vert.png")
    image = np.asarray(image_entrée)
    nb_lignes,nb_colonnes,_ = image.shape
    # On crée notre image de sortie sous forme de tableau numpy (ici on fait juste une copie
de l'image originale)
    image_sortie = np.copy(image)
    for ligne in range(nb_lignes):
        for col in range(nb_colonnes):
            r = image_sortie[ligne, col] * (1, 0, 0,1)
            g = image_sortie[ligne, col] * (0, 1, 0,1)
            b = image_sortie[ligne, col] * (0, 0, 1,1)
            if ((r[0] - 37) <= 110) and ((g[1] - 112) <= 110) and ((b[2] - 112) <= 110):
                image_sortie[ligne,col] = (255, 0, 0,255)
    # On sauvegarde les images pour pouvoir les afficher
    Image.fromarray(image).save("Info/image_entree.png")
    Image.fromarray(image_sortie).save("Info/image_sortie.png")
```