Assignment 4

Question 1

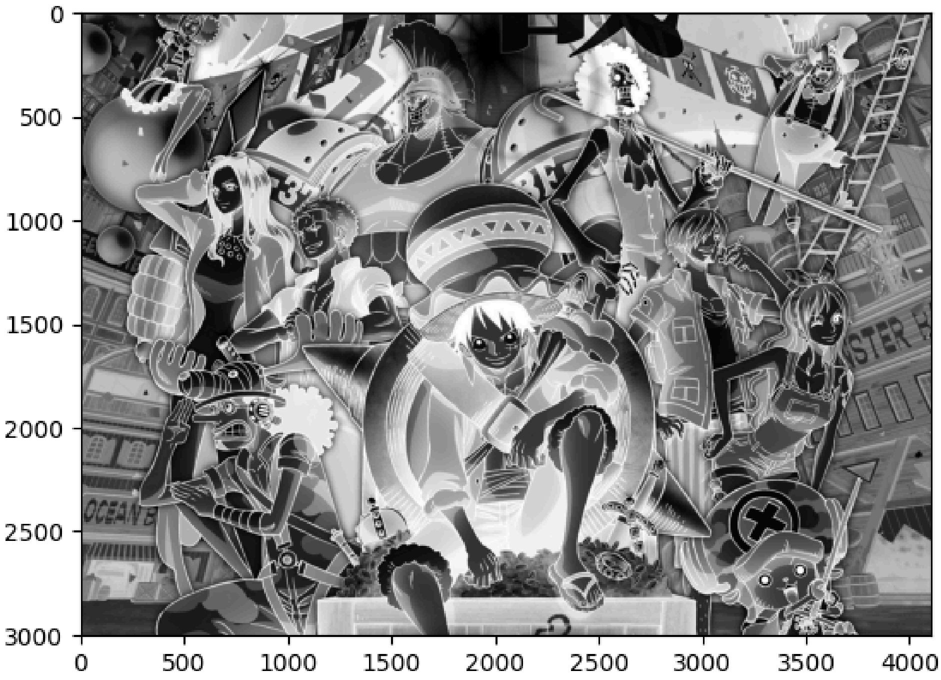
Apply the negative image transformation

Solution

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

img = cv2.imread("gray_image.jpg")

# Apply negative transformation
negative_img = np.max(img) - img
print(np.max(img))

cv2.imwrite("negative.jpg", negative_img)
plt.imshow(negative_img)
```

Output

```
255
<matplotlib.image.AxesImage at 0x7f6b80433bd0><Figure size 640x480 with 1 Axes>
```



Question 2

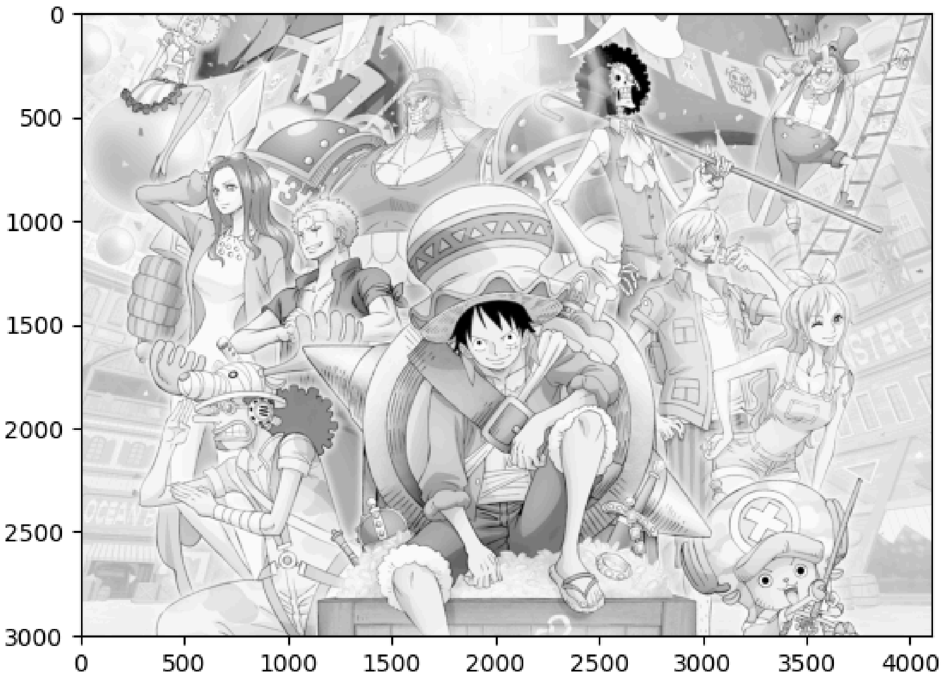
Apply the logarithmic image transformation

Solution

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

img = cv2.imread("gray_image.jpg")

img = img.astype(np.float32)

c = 255 / np.log(1 + np.max(img))
log_img = np.uint8(c * np.log1p(img))

cv2.imwrite("log_img.jpg", log_img)
plt.imshow(log_img)
```

Output

```
<matplotlib.image.AxesImage at 0x7f6b805ba210><Figure size 640x480 with 1 Axes>
```



Question 3

Make a custom 3x3 image and repeatedly apply the log transform

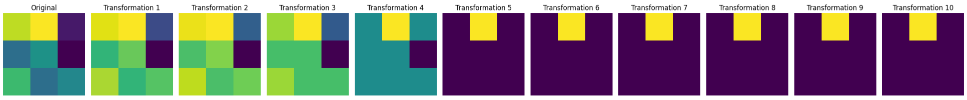
Solution

```
import numpy as np
import cv2
import matplotlib.pyplot as plt

new_img = (np.random.rand(3, 3) * 255).astype(np.uint8)


fig, axes = plt.subplots(1, 11, figsize=(2.2*(11), 4))
```

```
axes[0].imshow(new_img)
axes[0].set_title("Original")
axes[0].axis("off")

for i in range(10):
    c = 255 / np.log(1 + float(np.max(new_img)))
    log_img = np.uint8(c * np.log1p(new_img))

    axes[i+1].imshow(log_img)
    axes[i+1].set_title(f"Transformation {i+1}")
    axes[i+1].axis("off")

    new_img = log_img

plt.tight_layout()
plt.show()
```

Output

```
<Figure size 2420x400 with 11 Axes>
```



Question 3

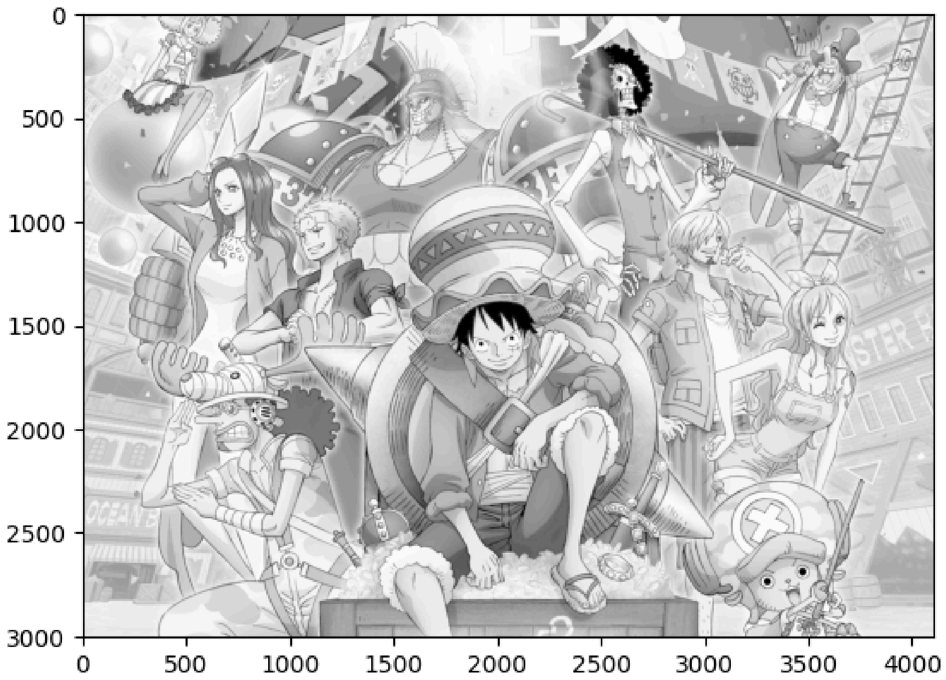
Apply the Power Law (Gamma Correction) image transformation

Solution

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

img = cv2.imread("gray_image.jpg")

gamma = 0.3
c = 1.0

norm_img = img / 255

power_log_img = c * np.power(norm_img, gamma)
plt.imshow(power_log_img)
```

Output

```
<matplotlib.image.AxesImage at 0x7f6b84534590><Figure size 640x480 with 1 Axes>
```



Question 5

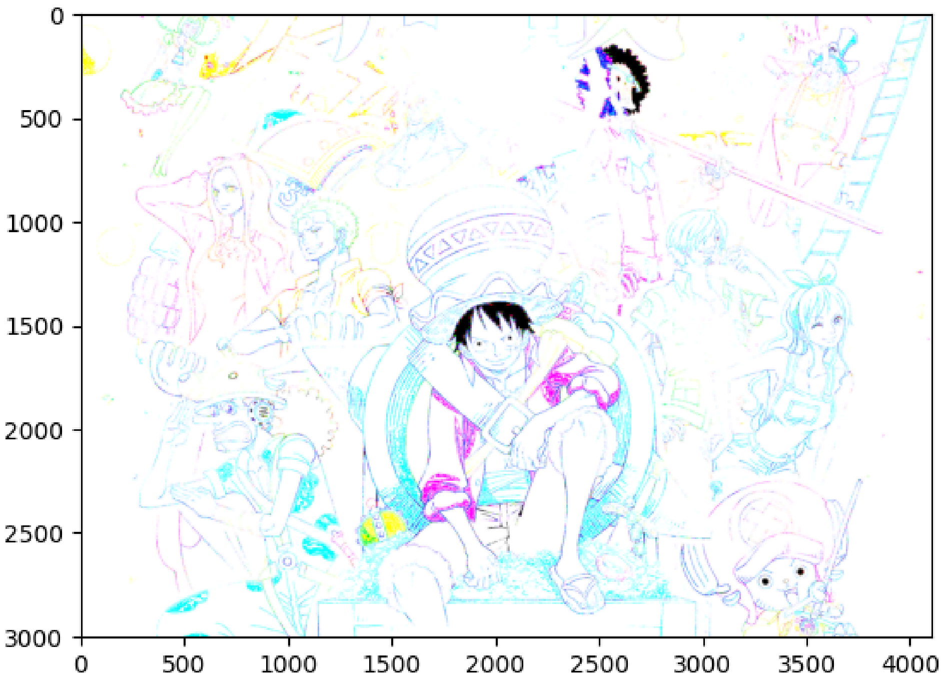
Apply the Contrast Stretching Transformation

Solution

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

img = cv2.imread("img1.jpg").astype(np.float32)

r1, r2 = 80, 120
s1, s2 = 0, 100

mask = (img >= r1) & (img <= r2)

img[mask] = ((img[mask] - 1) / (r2 - r1)) * (s2 - s1) + s1
img = np.clip(img, 0, 255).astype(np.float32)
plt.imshow(img)
```

Output

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or
[0..255] for integers). Got range [0.0..255.0].
<matplotlib.image.AxesImage at 0x7f6b803269d0><Figure size 640x480 with 1 Axes>
```


0
500
1000
1500
2000
2500
3000
0
500
1000
1500
2000
2500
3000
3500
4000

Solution