

```

1 import cv2
2 import numpy as np
3 import os
4
5 # PS1-3 (3) Gamma correction Function
6 def gamma_correction(image_input, gamma=1.0):
7     image_input = cv2.imread(user_input)
8     gamma_table = np.array([((i/255) ** gamma) *
255 for i in np.arange(0, 256)]).astype("uint8")
9     output_image = cv2.LUT(image_input, gamma_table
)
10     return output_image
11
12 # PS1-3 (1) User input file
13 user_input = input("Please name your input color
file: ")
14 file_directory = os.getcwd()
15 image_location = os.path.join(file_directory,
user_input)
16 if os.path.exists(image_location):
17     print(f"Your '{user_input}' image loaded
successfully.")
18     image_input = cv2.imread(user_input)
19     cv2.imshow('Input image', image_input)
20 else:
21     print(f"Error: unable to load your input image.
\nPlease make sure '{user_input}' is in the correct
directory.")
22     exit()
23
24 # PS1-3 (2) (4) User input gamma correction; Saving
final output files
25 user_gamma_value = input("Please indicate your
desire gamma correction value: ")
26 gamma_value = float(user_gamma_value)
27 if gamma_value > 0:
28     print(f"You choose '{gamma_value}' as your
gamma correction value.")
29     gamma_corrected_image = gamma_correction(
user_input, gamma=gamma_value)
30     cv2.imshow('Output image after gamma correction

```

```
30 ', gamma_corrected_image)
31     final_image = user_input.split('.')[0] + '
    _gcorrected.' + user_input.split('.')[-1]
32     cv2.imwrite( final_image, gamma_corrected_image
    )
33     cv2.waitKey(0)
34     cv2.destroyAllWindows('Circuit Color Image')
35 elif gamma_value < 0:
36     print(f"'{gamma_value}' is an invalid gamma
    correction value.")
37     exit()
38 else:
39     print(f"'{gamma_value}' is an invalid gamma
    correction value.")
40     exit()
41 exit()
```