```
from PIL import Image
import numpy as np

def encrypt_image(image_path, key):
    # Open the image
    img = Image.open(image_path)

    # Convert the image to a NumPy array
    img_array = np.array(img)

    # Ensure key has the same shape as img_array
    key = np.resize(key, img_array.shape)

    # Encrypt each pixel using XOR with the key
    encrypted_array = np.bitwise_xor(img_array, key)

    # Convert the encrypted array back to an image
    encrypted_img = Image.fromarray(encrypted_array)

    # Save the encrypted image
    encrypted_img.save("encrypted_image.png")
    print("Image encrypted successfully.")


def decrypt_image(encrypted_image_path, key):
    # Open the encrypted image
    encrypted_img = Image.open(encrypted_image_path)

    # Convert the encrypted image to a NumPy array
    encrypted_array = np.array(encrypted_img)

    # Ensure key has the same shape as encrypted_array
    key = np.resize(key, encrypted_array.shape)

    # Decrypt each pixel using XOR with the key
    decrypted_array = np.bitwise_xor(encrypted_array, ke

    # Convert the decrypted array back to an image
    decrypted_img = Image.fromarray(decrypted_array)

    # Save the decrypted image
    decrypted_img.save("decrypted_image.png")
    print("Image decrypted successfully.")


def main():
    print("Image Encryption and Decryption using Pixel M

    #image_path = 'C:\Users\HP\PRODIGY_INFOTECH\image1.p
    image_path = input("Enter the path to the image file
    # Generate a random key (you can use any integer as
    key = np.random.randint(0, 256, size=(3,), dtype=np.

    # Encrypt the image
```

Table of contents

+ Section

```
    encrypt_image(image_path, key)

    # Decrypt the image
    decrypt_image("encrypted_image.png", key)

if __name__ == "__main__":
    main()
```

```
Image Encryption and Decryption using Pixel Manipu
```