**ENCRYPTION:**

# try block to handle exception

try:

# take path of image as a input

path = input(r'Enter path of Image : ')

# taking encryption key as input

key = int(input('Enter Key for encryption of Image : '))

# print path of image file and encryption key that

# we are using

print('The path of file : ', path)

print('Key for encryption : ', key)

# open file for reading purpose

fin = open(path, 'rb')

# storing image data in variable "image"

image = fin.read()

fin.close()

# converting image into byte array to

# perform encryption easily on numeric data

image = bytearray(image)

# performing XOR operation on each value of bytearray

for index, values in enumerate(image):

image[index] = values ^ key

# opening file for writing purpose

fin = open(path, 'wb')

# writing encrypted data in image

fin.write(image)

fin.close()

print('Encryption Done...')

except Exception:

print('Error caught : ', Exception.\_\_name\_\_)

**DECRYPTION:**

# try block to handle the exception

try:

# take path of image as a input

path = input(r'Enter path of Image : ')

# taking decryption key as input

key = int(input('Enter Key for encryption of Image : '))

# print path of image file and decryption key that we are using

print('The path of file : ', path)

print('Note : Encryption key and Decryption key must be same.')

print('Key for Decryption : ', key)

# open file for reading purpose

fin = open(path, 'rb')

# storing image data in variable "image"

image = fin.read()

fin.close()

# converting image into byte array to perform decryption easily on numeric data

image = bytearray(image)

# performing XOR operation on each value of bytearray

for index, values in enumerate(image):

image[index] = values ^ key

# opening file for writing purpose

fin = open(path, 'wb')

# writing decryption data in image

fin.write(image)

fin.close()

print('Decryption Done...')

except Exception:

print('Error caught : ', Exception.\_\_name\_\_)