Image Encryption Tool

def encrypt\_decrypt\_bmp(input\_path, output\_path, key):

try:

with open(input\_path, 'rb') as f:

data = bytearray(f.read())

for i in range(54, len(data)):

data[i] ^= key # XOR encryption

with open(output\_path, 'wb') as f:

f.write(data)

print(f"Output saved to {output\_path}")

except FileNotFoundError:

print("File not found.")

except Exception as e:

print(f"Error: {e}")

def main():

print("=== BMP Image XOR Encryption Tool ===")

choice = input("Do you want to (E)ncrypt or (D)ecrypt? ").strip().upper()

if choice not in ('E', 'D'):

print("Invalid choice. Use 'E' or 'D'.")

return

input\_path = input("Enter path to BMP file: ").strip()

output\_path = input("Enter path to save output BMP: ").strip()

try:

key = int(input("Enter key (0–255): "))

if not 0 <= key <= 255:

raise ValueError

except ValueError:

print("Key must be an integer between 0 and 255.")

return

encrypt\_decrypt\_bmp(input\_path, output\_path, key)

if \_\_name\_\_ == "\_\_main\_\_":

main()

Output:

