import base64  
  
def extract\_key\_from\_bmp(bmp\_path):  
 with open(bmp\_path, 'rb') as f:  
 d = bytearray(f.read())  
  
 # 读取BMP文件头信息  
 pixel\_offset = int.from\_bytes(d[10:14], 'little')  
  
 # 提取LSB中的二进制数据（前256位）  
 secret\_bits = []  
 for i in range(pixel\_offset, len(d)):  
 if len(secret\_bits) >= 256:  
 break  
 secret\_bits.append(d[i] & 0x01) # 取最低位  
  
 # 将二进制位转换为字节  
 key\_bytes = []  
 for i in range(0, 256, 8):  
 byte\_bits = secret\_bits[i:i + 8]  
 byte\_value = int(''.join(map(str, byte\_bits)), 2)  
 key\_bytes.append(byte\_value)  
  
 return bytes(key\_bytes)  
  
  
def decrypt\_flag(encrypted\_b85\_path, key):  
 with open(encrypted\_b85\_path, 'rb') as f:  
 encrypted\_data = base64.b85decode(f.read())  
  
 # 异或解密  
 decrypted = bytes([encrypted\_data[i] ^ key[i % len(key)]  
 for i in range(len(encrypted\_data))])  
 return decrypted.decode()  
  
  
# 主流程  
if \_\_name\_\_ == "\_\_main\_\_":  
 # 1. 从secret.bmp提取密钥c  
 c = extract\_key\_from\_bmp("secret.bmp")  
  
 # 2. 解密encrypted.b85  
 flag = decrypt\_flag("encrypted.b85", c)  
 print("Flag:", flag)

wrhklm{f9a63ded21b9fbf6025ff3f50351d9ef31078928}