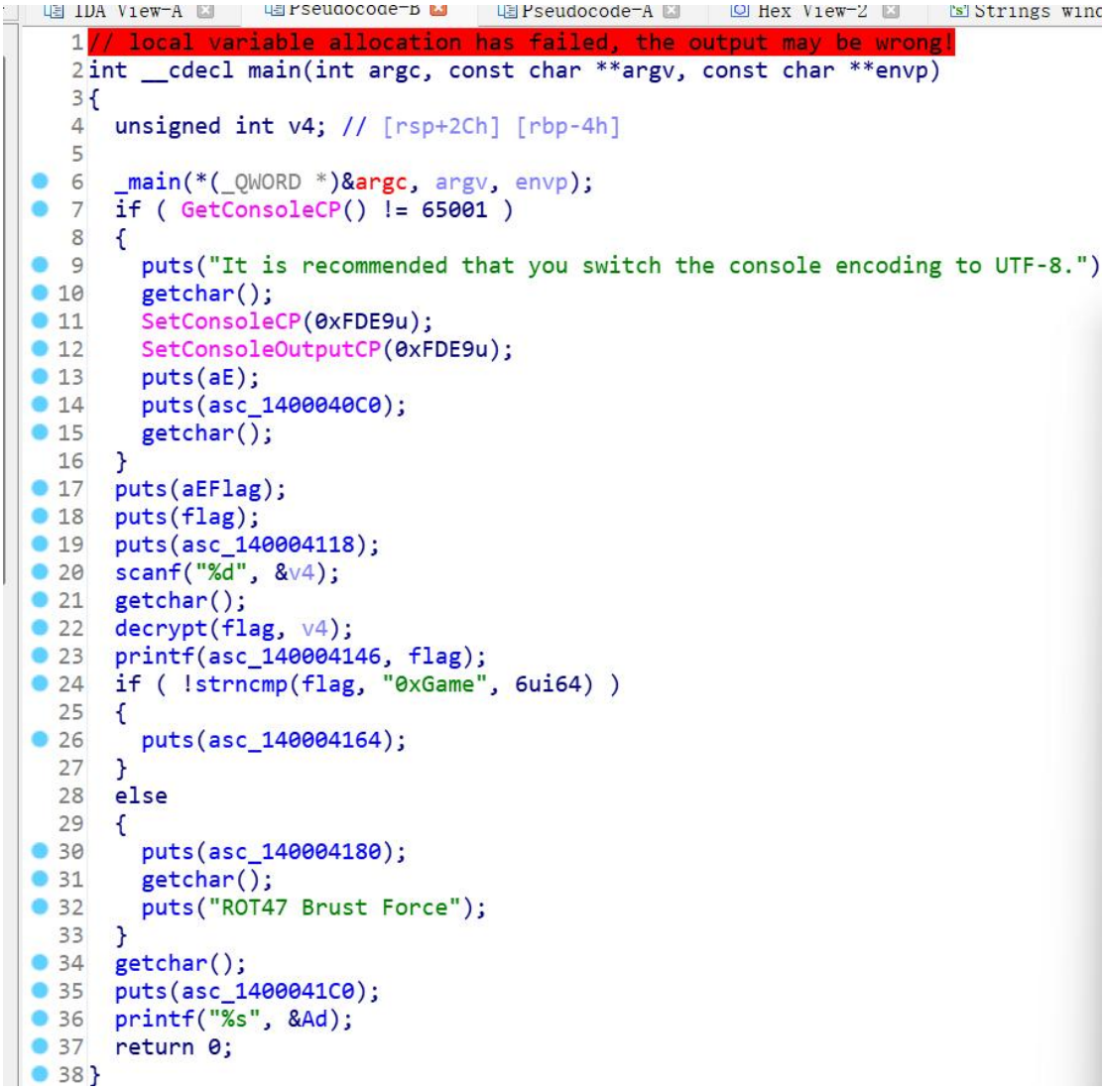
0xgame

Reverse

signIn 拖入 ida 直接就有

signIn2



```
// local variable allocation has failed, the output may be wrong!
int __cdecl main(int argc, const char **argv, const char **envp)
{
  unsigned int v4; // [rsp+2Ch] [rbp-4h]

  _main(*(_QWORD *)&argc, argv, envp);
  if ( GetConsoleCP() != 65001 )
  {
    puts("It is recommended that you switch the console encoding to UTF-8.");
    getchar();
    SetConsoleCP(0xFDE9u);
    SetConsoleOutputCP(0xFDE9u);
    puts(aE);
    puts(asc_1400040C0);
    getchar();
  }
  puts(aEFlag);
  puts(flag);
  puts(asc_140004118);
  scanf("%d", &v4);
  getchar();
  decrypt(flag, v4);
  printf(asc_140004146, flag);
  if ( !strncmp(flag, "0xGame", 6ui64) )
  {
    puts(asc_140004164);
  }
  else
  {
    puts(asc_140004180);
    getchar();
    puts("ROT47 Brust Force");
  }
  getchar();
  puts(asc_1400041C0);
  printf("%s", &Ad);
  return 0;
}
```

```
.data:00000001400035E0                 public flag
.data:00000001400035E0 ; char flag[]
.data:00000001400035E0 flag            db '@*Wq}u-guAs@}CoBo*yq!*y~*yuo##oA@F@DDIE@I/',0
.data:00000001400035E0                                         ; DATA XREF: main+7F↑o
.data:00000001400035E0                                         ; main+B8↑o ...
.data:00000001400035E0                 align 20h
```

直接使用 rot47 并不能解密出正确 flag，尝试其他偏移量，寻找以 0xgame 开头的的结果

```
C:\WINDOWS\system32\cmd.
解密结果：pZ)COG]9GqEpOsArAZKCQZKPZKGASSAqpvpttyupy_
>>> def decrypt(text, offset):
...     result = []
...     for char in text:
...         v5 = ord(char)
...         if (v5 - 32) > 0x5E:
...             result.append(char)
...         else:
...             new_char = (v5 - 32 - offset % 0x5E + 95) % 0x5E + 32
...             result.append(chr(new_char))
...     return ''.join(result)
...
... encoded_text = "@*Wq}u-guAs@}CoBo*yq!*y~*yuo##oA@F@DDIE@I/"
...
... # 尝试所有可能的偏移量（0-93）
... print("尝试所有偏移量，寻找以'0xgame'开头的结果：")
... for offset in range(94):
...     result = decrypt(encoded_text, offset)
...     if result.startswith("0xgame") or "0xgame" in result.lower():
...         print(f"✅ 找到正确偏移量：{offset}")
...         print(f"解密结果：{result}")
...         break
...     # 也可以检查其他常见开头
...     if result.startswith(("flag", "0xGame", "FLAG")):
...         print(f"✅ 可能正确（偏移量{offset}）：{result}")
...
尝试所有偏移量，寻找以'0xgame'开头的结果：
✅ 找到正确偏移量：17
解密结果：0xGame{Welc0m3_2_xiaoxinxie_qq_1060449509}
>>>
```

BaseUpx 题目已经提示的非常明白，先脱壳，

```
D:\upx-5.0.2-win64>upx -d "C:\Users\1\Downloads\BaseUpx\BaseUpx.exe"
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2025
UPX 5.0.2       Markus Oberhumer, Laszlo Molnar & John Reiser   Jul 20th 2025

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
     60664 <-     43768   72.15%    win64/pe     BaseUpx.exe

Unpacked 1 file.
```

然后拖入 ida 可看到经过 base64 编码的如下字符

```
.rdata:0000000000405000                 ;org 405000h
.rdata:0000000000405000 aAbcdefghijklmn db 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/',0
.rdata:0000000000405000                                         ; DATA XREF: .data:base64_charsto
.rdata:0000000000405041                 align 8
.rdata:0000000000405048 aMhhhyw1le1cwd1 db 'MHhHYW1le1cwd191XzRyM183aDNfRzBkXzBmX3VweCZiNHMzNjRfRDNzMWdufQ==',0
.rdata:0000000000405048                                         ; DATA XREF: .data:strto
.rdata:0000000000405089 ; char Str[]
.rdata:0000000000405089 Str             db 'Input your flag:',0 ; DATA XREF: main+Dto
.rdata:000000000040509A ; char Control[]
```

解码即得到 flag

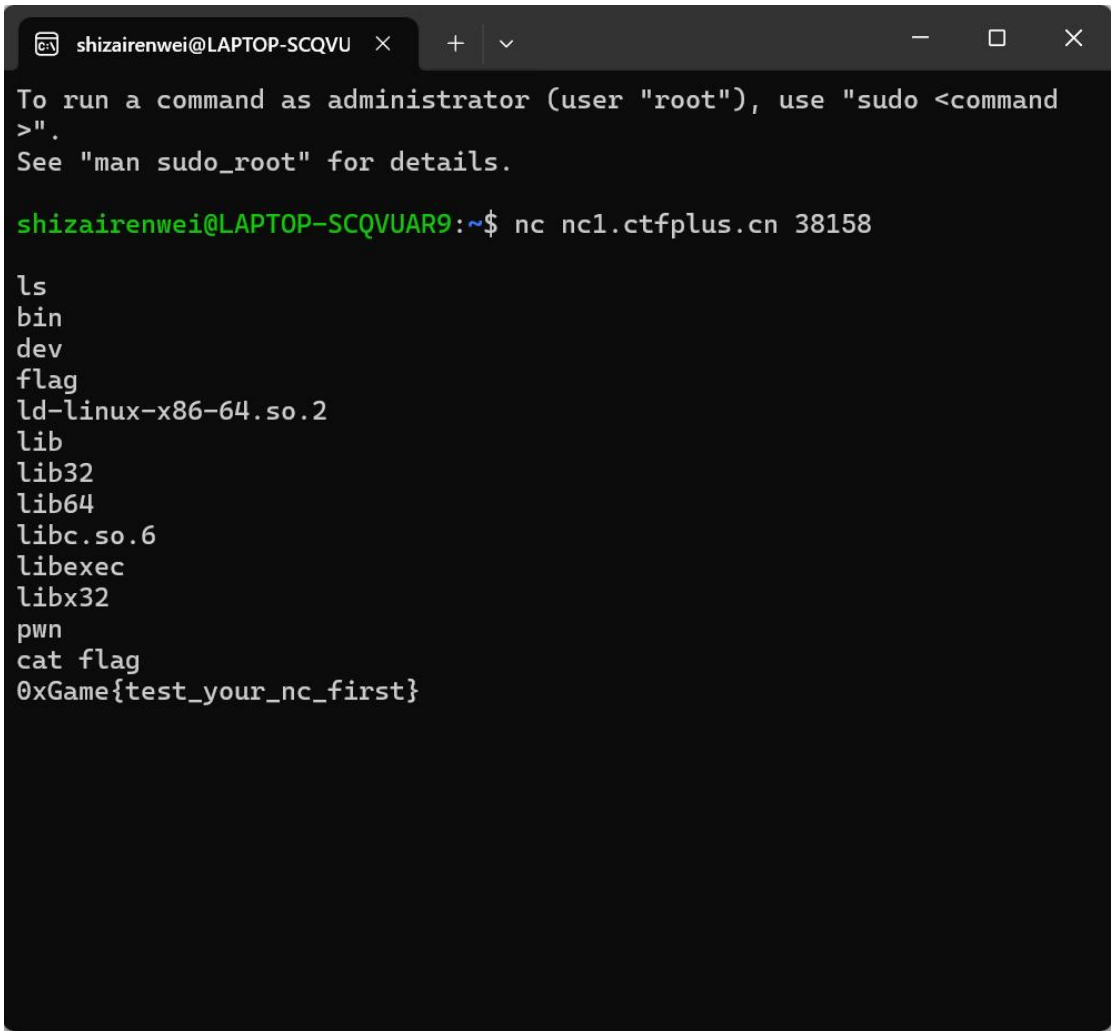
MHhHYW1le1cwd191XzRyM183aDNfRzBkXzBmX3VweCZiNHMzNjRfRDNzMWdufQ==

字符编码： UTF-8 解码过滤非 Base64 字符 Base64 编码

0xGame{W0w_u_4r3_7h3_G0d_0f_upx&b4s364_D3s1gn}

pwn

test your nc



ls 和 cat flag 组合技

NewStarCTF2025

XOr 附件找不到了，不过这是两轮异或加密，且每轮密钥是循环的，第一轮是 0x14,0x11,0x45 循环，第二轮是 19，19，81 循环，用密文异或同样密钥即可

```
>>> # 正确的密钥
... key1 = [0x14, 0x11, 0x45]
... key2 = [19, 19, 81]
...
... # 密文
... ciphertext = b"anu`ym7wKLl$P]v3q%D]lHpi"
...
... flag = []
... for i in range(len(ciphertext)):
...     # 解密：先异或 key2，再异或 key1
...     byte = ciphertext[i] ^ key2[i % 3]
...     byte = byte ^ key1[i % 3]
...     flag.append(byte)
...
... print("解密结果:", bytes(flag))
...
解密结果： b'flag{y0u_Kn0W_b4s1C_xOr}'
>>>
```