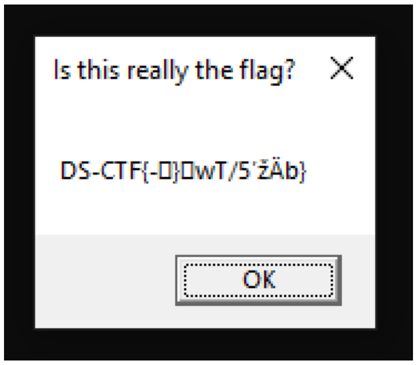
Our honeypot caught something...

Task:

Find the flag! Remember to add "DS-CTF{flag_here}" in case it is missing

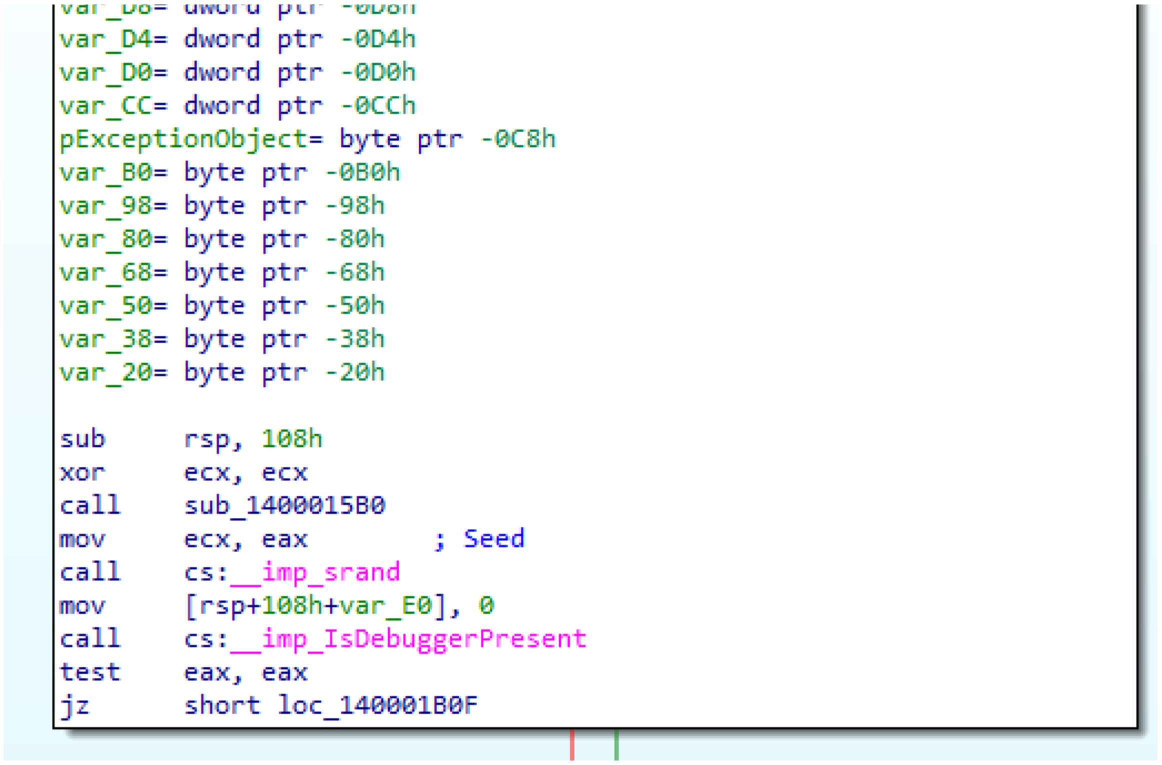
1. Running the binary gives you an invalid flag.



Is this really the flag?

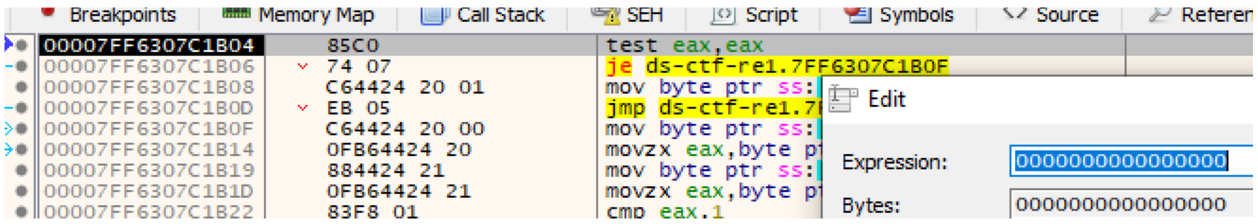
```
DS-CTF{-□}□wT/5´žÄb}
```

2. Checking with IDA we can see that the binary is checking for debuggers so when debugging we need to get around that. By simply changing the return value we get around that check



```
var_D4= dword ptr -0D4h
var_D0= dword ptr -0D0h
var_CC= dword ptr -0CCh
pExceptionObject= byte ptr -0C8h
var_B0= byte ptr -0B0h
var_98= byte ptr -98h
var_80= byte ptr -80h
var_68= byte ptr -68h
var_50= byte ptr -50h
var_38= byte ptr -38h
var_20= byte ptr -20h

sub     rsp, 108h
xor     ecx, ecx
call    sub_1400015B0
mov     ecx, eax        ; Seed
call    cs:__imp_srand
mov     [rsp+108h+var_E0], 0
call    cs:__imp_IsDebuggerPresent
test    eax, eax
jz      short loc_140001B0F
```

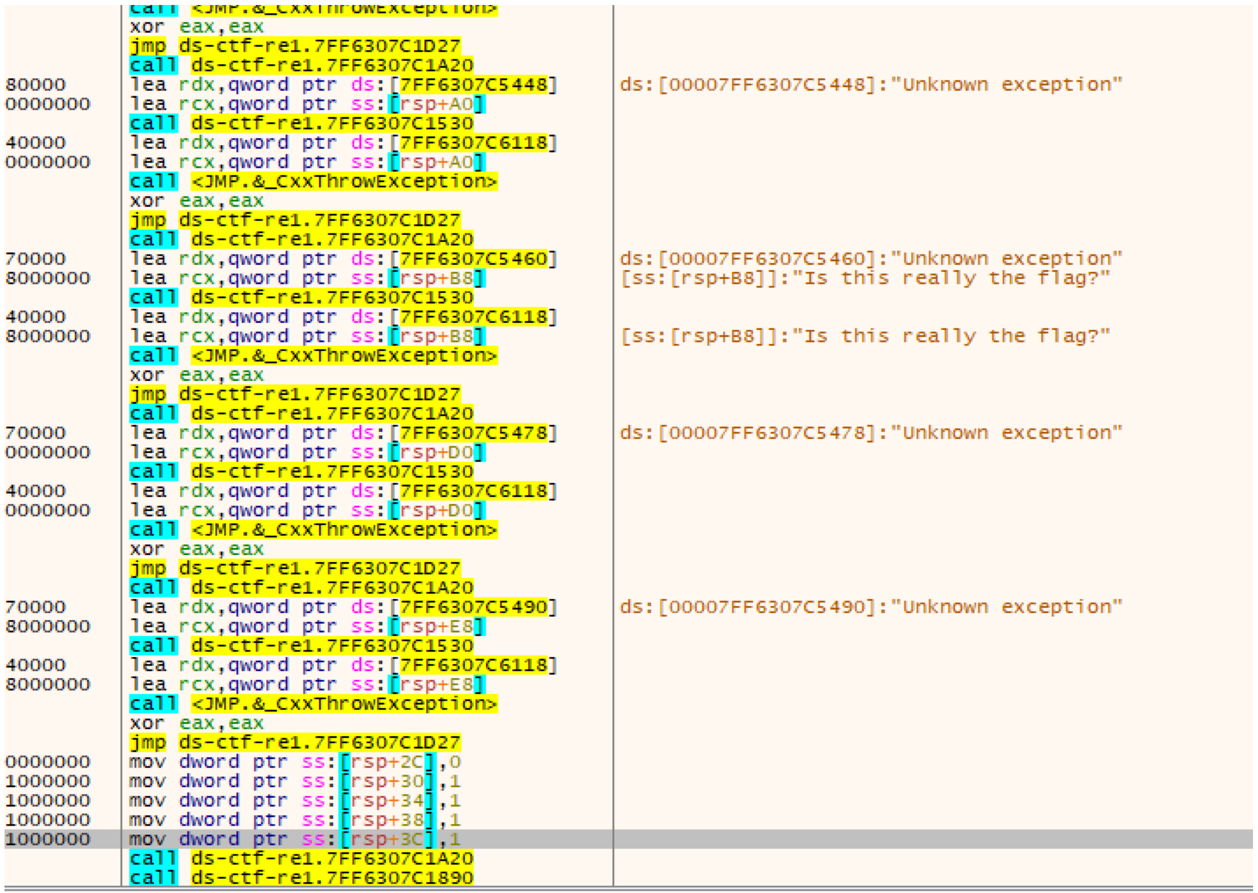
3. 

```
00007FF6307C1B04   85C0            test eax,eax
00007FF6307C1B06   74 07           je ds-ctf-re1.7FF6307C1B0F
00007FF6307C1B08   C64424 20 01    mov byte ptr ss:[
00007FF6307C1B0D   EB 05           jmp ds-ctf-re1.7F
00007FF6307C1B0F   C64424 20 00    mov byte ptr ss:[
00007FF6307C1B14   0FB64424 20     movzx eax,byte pt
00007FF6307C1B19   884424 21       mov byte ptr ss:[
00007FF6307C1B1D   0FB64424 21     movzx eax,byte pt
00007FF6307C1B22   83F8 01         cmp eax,1
```

Expression: 0000000000000000

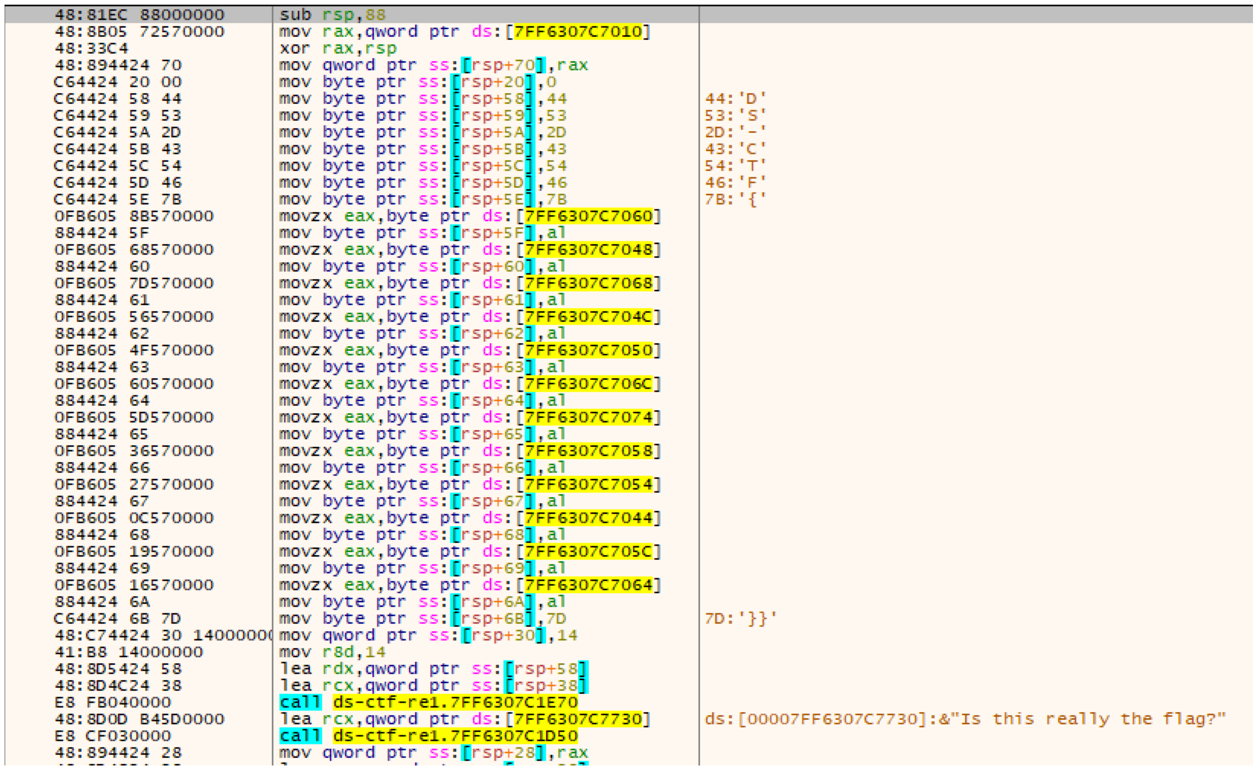
Bytes: 0000000000000000

4. With the debugger-check handled we can step through the code until we get to a larger switch-case which includes the seemingly random flag



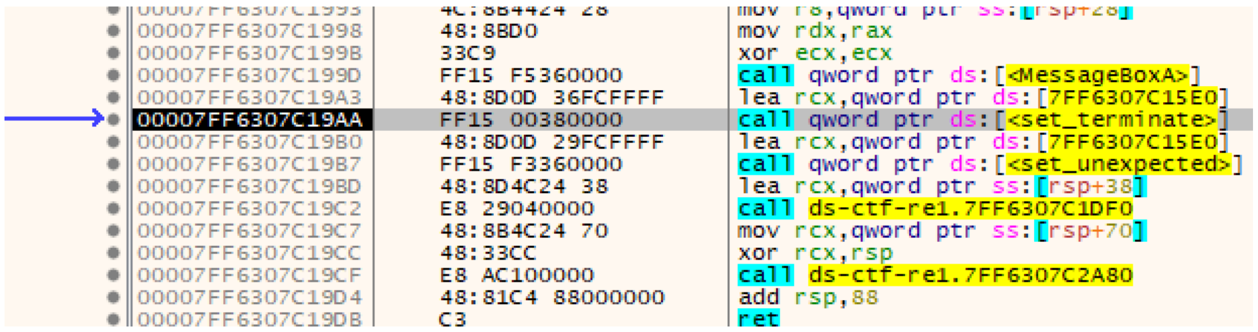
```
          call <JMP.&_CxxThrowException>
          xor eax,eax
          jmp ds-ctf-re1.7FF6307C1D27
          call ds-ctf-re1.7FF6307C1A20
80000     lea rdx,qword ptr ds:[7FF6307C5448]     ds:[00007FF6307C5448]:"Unknown exception"
0000000   lea rcx,qword ptr ss:[rsp+A0]
          call ds-ctf-re1.7FF6307C1530
40000     lea rdx,qword ptr ds:[7FF6307C6118]
0000000   lea rcx,qword ptr ss:[rsp+A0]
          call <JMP.&_CxxThrowException>
          xor eax,eax
          jmp ds-ctf-re1.7FF6307C1D27
          call ds-ctf-re1.7FF6307C1A20
70000     lea rdx,qword ptr ds:[7FF6307C5460]     ds:[00007FF6307C5460]:"Unknown exception"
8000000   lea rcx,qword ptr ss:[rsp+B8]           [ss:[rsp+B8]]:"Is this really the flag?"
          call ds-ctf-re1.7FF6307C1530
40000     lea rdx,qword ptr ds:[7FF6307C6118]
8000000   lea rcx,qword ptr ss:[rsp+B8]           [ss:[rsp+B8]]:"Is this really the flag?"
          call <JMP.&_CxxThrowException>
          xor eax,eax
          jmp ds-ctf-re1.7FF6307C1D27
          call ds-ctf-re1.7FF6307C1A20
70000     lea rdx,qword ptr ds:[7FF6307C5478]     ds:[00007FF6307C5478]:"Unknown exception"
0000000   lea rcx,qword ptr ss:[rsp+D0]
          call ds-ctf-re1.7FF6307C1530
40000     lea rdx,qword ptr ds:[7FF6307C6118]
0000000   lea rcx,qword ptr ss:[rsp+D0]
          call <JMP.&_CxxThrowException>
          xor eax,eax
          jmp ds-ctf-re1.7FF6307C1D27
          call ds-ctf-re1.7FF6307C1A20
70000     lea rdx,qword ptr ds:[7FF6307C5490]     ds:[00007FF6307C5490]:"Unknown exception"
8000000   lea rcx,qword ptr ss:[rsp+E8]
          call ds-ctf-re1.7FF6307C1530
40000     lea rdx,qword ptr ds:[7FF6307C6118]
8000000   lea rcx,qword ptr ss:[rsp+E8]
          call <JMP.&_CxxThrowException>
          xor eax,eax
          jmp ds-ctf-re1.7FF6307C1D27
0000000   mov dword ptr ss:[rsp+2C],0
1000000   mov dword ptr ss:[rsp+30],1
1000000   mov dword ptr ss:[rsp+34],1
1000000   mov dword ptr ss:[rsp+38],1
1000000   mov dword ptr ss:[rsp+3C],1
          call ds-ctf-re1.7FF6307C1A20
          call ds-ctf-re1.7FF6307C1890
```

5. As those are seemingly not important we take a closer look at the 2 functions at the bottom
   The second function seems to write out part of the flag



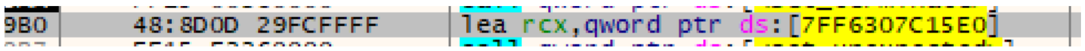
```
48:81EC 88000000          sub rsp,88
48:8B05 72570000          mov rax,qword ptr ds:[7FF6307C7010]
48:33C4                   xor rax,rsp
48:894424 70              mov qword ptr ss:[rsp+70],rax
C64424 20 00              mov byte ptr ss:[rsp+20],0
C64424 58 44              mov byte ptr ss:[rsp+58],44             44:'D'
C64424 59 53              mov byte ptr ss:[rsp+59],53             53:'S'
C64424 5A 2D              mov byte ptr ss:[rsp+5A],2D             2D:'-'
C64424 5B 43              mov byte ptr ss:[rsp+5B],43             43:'C'
C64424 5C 54              mov byte ptr ss:[rsp+5C],54             54:'T'
C64424 5D 46              mov byte ptr ss:[rsp+5D],46             46:'F'
C64424 5E 7B              mov byte ptr ss:[rsp+5E],7B             7B:'{'
0FB605 8B570000           movzx eax,byte ptr ds:[7FF6307C7060]
884424 5F                 mov byte ptr ss:[rsp+5F],al
0FB605 68570000           movzx eax,byte ptr ds:[7FF6307C7048]
884424 60                 mov byte ptr ss:[rsp+60],al
0FB605 7D570000           movzx eax,byte ptr ds:[7FF6307C7068]
884424 61                 mov byte ptr ss:[rsp+61],al
0FB605 56570000           movzx eax,byte ptr ds:[7FF6307C704C]
884424 62                 mov byte ptr ss:[rsp+62],al
0FB605 4F570000           movzx eax,byte ptr ds:[7FF6307C7050]
884424 63                 mov byte ptr ss:[rsp+63],al
0FB605 60570000           movzx eax,byte ptr ds:[7FF6307C706C]
884424 64                 mov byte ptr ss:[rsp+64],al
0FB605 5D570000           movzx eax,byte ptr ds:[7FF6307C7074]
884424 65                 mov byte ptr ss:[rsp+65],al
0FB605 36570000           movzx eax,byte ptr ds:[7FF6307C7058]
884424 66                 mov byte ptr ss:[rsp+66],al
0FB605 27570000           movzx eax,byte ptr ds:[7FF6307C7054]
884424 67                 mov byte ptr ss:[rsp+67],al
0FB605 0C570000           movzx eax,byte ptr ds:[7FF6307C7044]
884424 68                 mov byte ptr ss:[rsp+68],al
0FB605 19570000           movzx eax,byte ptr ds:[7FF6307C705C]
884424 69                 mov byte ptr ss:[rsp+69],al
0FB605 16570000           movzx eax,byte ptr ds:[7FF6307C7064]
884424 6A                 mov byte ptr ss:[rsp+6A],al
C64424 6B 7D              mov byte ptr ss:[rsp+6B],7D             7D:'}}'
48:C74424 30 14000000     mov qword ptr ss:[rsp+30],14
41:B8 14000000            mov r8d,14
48:8D5424 58              lea rdx,qword ptr ss:[rsp+58]
48:8D4C24 38              lea rcx,qword ptr ss:[rsp+38]
E8 FB040000               call ds-ctf-re1.7FF6307C1E70
48:8D0D B45D0000          lea rcx,qword ptr ds:[7FF6307C7730]     ds:[00007FF6307C7730]:&"Is this really the flag?"
E8 CF030000               call ds-ctf-re1.7FF6307C1D50
48:894424 28              mov qword ptr ss:[rsp+28],rax
```

6. After the messagebox with the “false” flag we can see 2 calls to set_terminate and set_unexpected. The binary is setting up its own exception handler



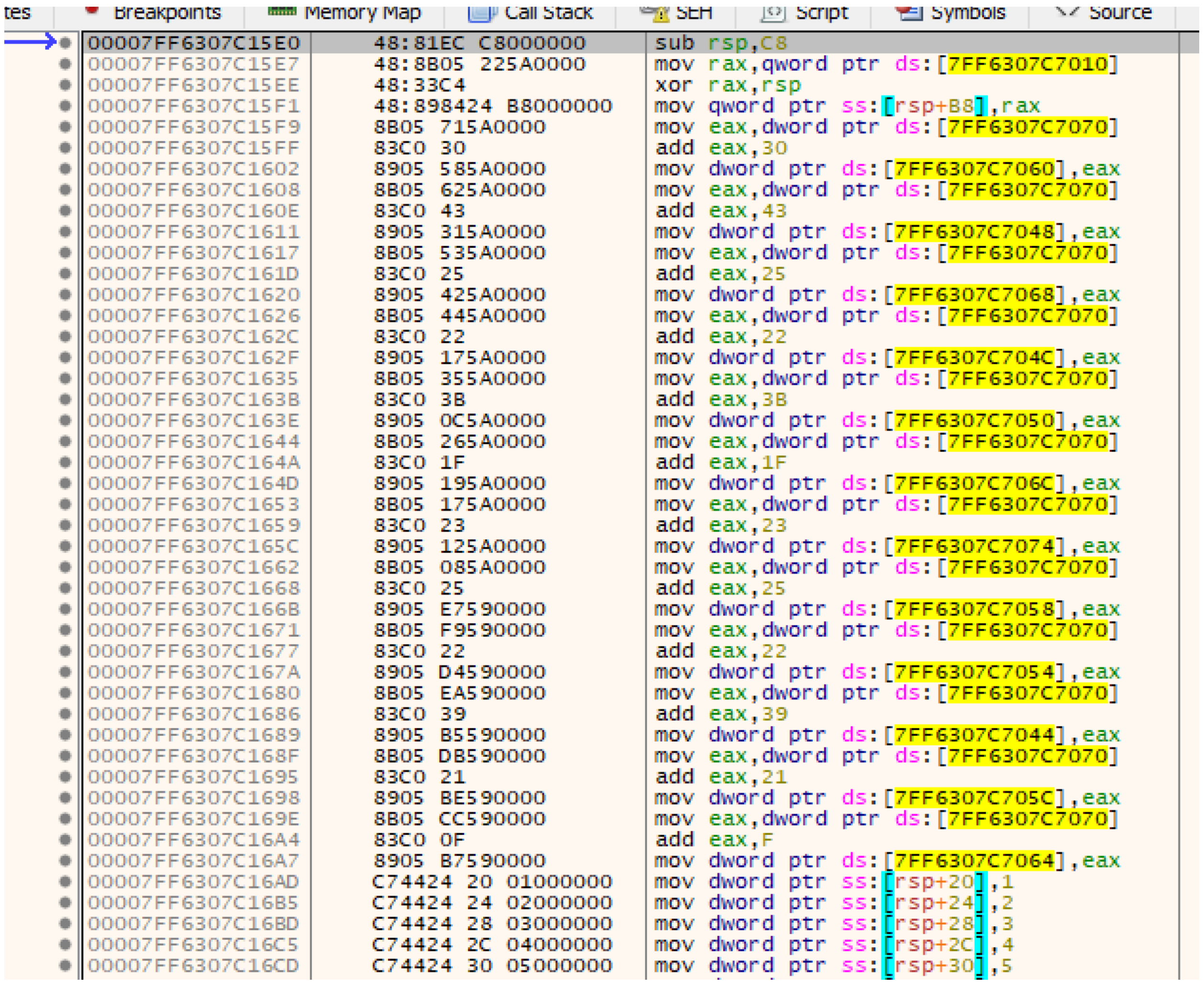
```
00007FF6307C1993   4C:8B4424 28     mov r8,qword ptr ss:[rsp+28]
00007FF6307C1998   48:8BD0          mov rdx,rax
00007FF6307C199B   33C9             xor ecx,ecx
00007FF6307C199D   FF15 F5360000    call qword ptr ds:[<MessageBoxA>]
00007FF6307C19A3   48:8D0D 36FCFFFF lea rcx,qword ptr ds:[7FF6307C15E0]
00007FF6307C19AA   FF15 00380000    call qword ptr ds:[<set_terminate>]
00007FF6307C19B0   48:8D0D 29FCFFFF lea rcx,qword ptr ds:[7FF6307C15E0]
00007FF6307C19B7   FF15 F3360000    call qword ptr ds:[<set_unexpected>]
00007FF6307C19BD   48:8D4C24 38     lea rcx,qword ptr ss:[rsp+38]
00007FF6307C19C2   E8 29040000      call ds-ctf-re1.7FF6307C1DF0
00007FF6307C19C7   48:8B4C24 70     mov rcx,qword ptr ss:[rsp+70]
00007FF6307C19CC   48:33CC          xor rcx,rsp
00007FF6307C19CF   E8 AC100000      call ds-ctf-re1.7FF6307C2A80
00007FF6307C19D4   48:81C4 88000000 add rsp,88
00007FF6307C19DB   C3               ret
```

7. To see which function is supposed to be called we can check the value in lea rcx…



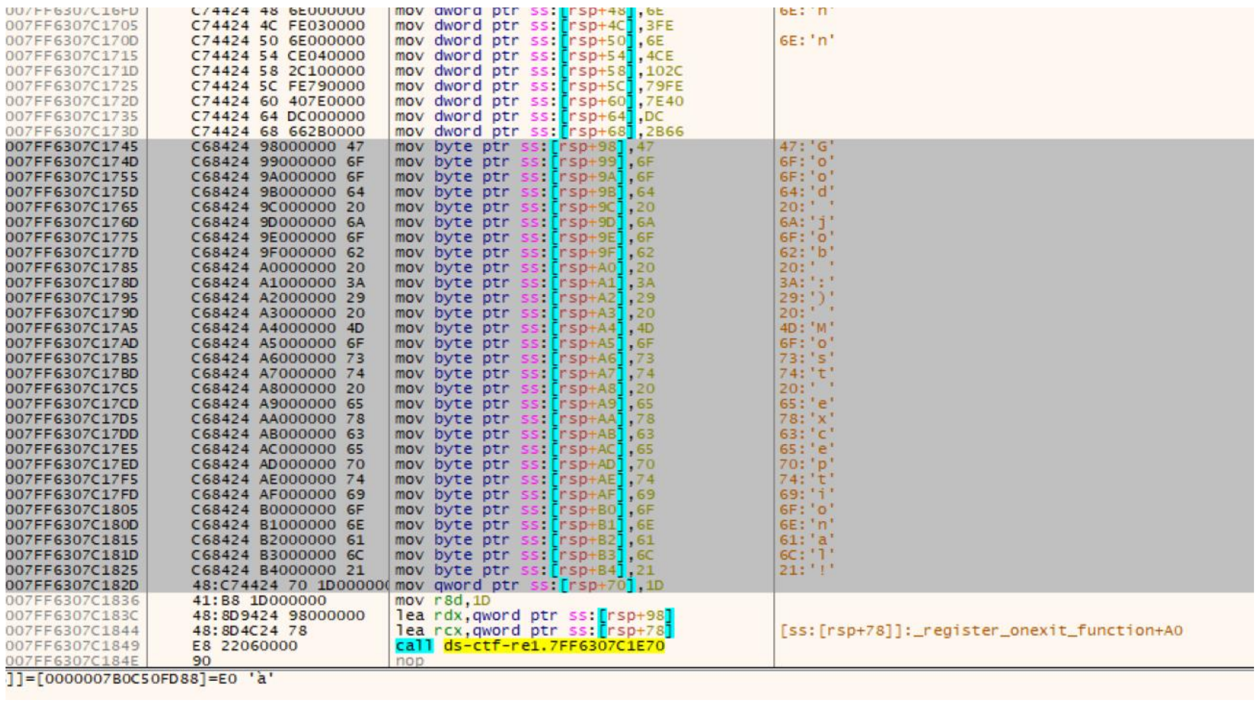
```
9B0   48:8D0D 29FCFFFF   lea rcx,qword ptr ds:[7FF6307C15E0]
```

8. Following the code we will see a lot of xor operations



```
00007FF6307C15E0   48:81EC C8000000        sub rsp,C8
00007FF6307C15E7   48:8B05 225A0000        mov rax,qword ptr ds:[7FF6307C7010]
00007FF6307C15EE   48:33C4                 xor rax,rsp
00007FF6307C15F1   48:898424 B8000000      mov qword ptr ss:[rsp+B8],rax
00007FF6307C15F9   8B05 715A0000           mov eax,dword ptr ds:[7FF6307C7070]
00007FF6307C15FF   83C0 30                 add eax,30
00007FF6307C1602   8905 585A0000           mov dword ptr ds:[7FF6307C7060],eax
00007FF6307C1608   8B05 625A0000           mov eax,dword ptr ds:[7FF6307C7070]
00007FF6307C160E   83C0 43                 add eax,43
00007FF6307C1611   8905 315A0000           mov dword ptr ds:[7FF6307C7048],eax
00007FF6307C1617   8B05 535A0000           mov eax,dword ptr ds:[7FF6307C7070]
00007FF6307C161D   83C0 25                 add eax,25
00007FF6307C1620   8905 425A0000           mov dword ptr ds:[7FF6307C7068],eax
00007FF6307C1626   8B05 445A0000           mov eax,dword ptr ds:[7FF6307C7070]
00007FF6307C162C   83C0 22                 add eax,22
00007FF6307C162F   8905 175A0000           mov dword ptr ds:[7FF6307C704C],eax
00007FF6307C1635   8B05 355A0000           mov eax,dword ptr ds:[7FF6307C7070]
00007FF6307C163B   83C0 3B                 add eax,3B
00007FF6307C163E   8905 0C5A0000           mov dword ptr ds:[7FF6307C7050],eax
00007FF6307C1644   8B05 265A0000           mov eax,dword ptr ds:[7FF6307C7070]
00007FF6307C164A   83C0 1F                 add eax,1F
00007FF6307C164D   8905 195A0000           mov dword ptr ds:[7FF6307C706C],eax
00007FF6307C1653   8B05 175A0000           mov eax,dword ptr ds:[7FF6307C7070]
00007FF6307C1659   83C0 23                 add eax,23
00007FF6307C165C   8905 125A0000           mov dword ptr ds:[7FF6307C7074],eax
00007FF6307C1662   8B05 085A0000           mov eax,dword ptr ds:[7FF6307C7070]
00007FF6307C1668   83C0 25                 add eax,25
00007FF6307C166B   8905 E7590000           mov dword ptr ds:[7FF6307C7058],eax
00007FF6307C1671   8B05 F9590000           mov eax,dword ptr ds:[7FF6307C7070]
00007FF6307C1677   83C0 22                 add eax,22
00007FF6307C167A   8905 D4590000           mov dword ptr ds:[7FF6307C7054],eax
00007FF6307C1680   8B05 EA590000           mov eax,dword ptr ds:[7FF6307C7070]
00007FF6307C1686   83C0 39                 add eax,39
00007FF6307C1689   8905 B5590000           mov dword ptr ds:[7FF6307C7044],eax
00007FF6307C168F   8B05 DB590000           mov eax,dword ptr ds:[7FF6307C7070]
00007FF6307C1695   83C0 21                 add eax,21
00007FF6307C1698   8905 BE590000           mov dword ptr ds:[7FF6307C705C],eax
00007FF6307C169E   8B05 CC590000           mov eax,dword ptr ds:[7FF6307C7070]
00007FF6307C16A4   83C0 0F                 add eax,F
00007FF6307C16A7   8905 B7590000           mov dword ptr ds:[7FF6307C7064],eax
00007FF6307C16AD   C74424 20 01000000      mov dword ptr ss:[rsp+20],1
00007FF6307C16B5   C74424 24 02000000      mov dword ptr ss:[rsp+24],2
00007FF6307C16BD   C74424 28 03000000      mov dword ptr ss:[rsp+28],3
00007FF6307C16C5   C74424 2C 04000000      mov dword ptr ss:[rsp+2C],4
00007FF6307C16CD   C74424 30 05000000      mov dword ptr ss:[rsp+30],5
```

9. Scrolling down we can see the actual flag



```
007FF6307C16FD   C74424 48 6E000000     mov dword ptr ss:[rsp+48],6E        6E:'n'
007FF6307C1705   C74424 4C FE030000     mov dword ptr ss:[rsp+4C],3FE
007FF6307C170D   C74424 50 6E000000     mov dword ptr ss:[rsp+50],6E        6E:'n'
007FF6307C1715   C74424 54 CE040000     mov dword ptr ss:[rsp+54],4CE
007FF6307C171D   C74424 58 2C100000     mov dword ptr ss:[rsp+58],102C
007FF6307C1725   C74424 5C FE790000     mov dword ptr ss:[rsp+5C],79FE
007FF6307C172D   C74424 60 407E0000     mov dword ptr ss:[rsp+60],7E40
007FF6307C1735   C74424 64 DC000000     mov dword ptr ss:[rsp+64],DC
007FF6307C173D   C74424 68 662B0000     mov dword ptr ss:[rsp+68],2B66
007FF6307C1745   C68424 98000000 47     mov byte ptr ss:[rsp+98],47         47:'G'
007FF6307C174D   C68424 99000000 6F     mov byte ptr ss:[rsp+99],6F         6F:'o'
007FF6307C1755   C68424 9A000000 6F     mov byte ptr ss:[rsp+9A],6F         6F:'o'
007FF6307C175D   C68424 9B000000 64     mov byte ptr ss:[rsp+9B],64         64:'d'
007FF6307C1765   C68424 9C000000 20     mov byte ptr ss:[rsp+9C],20         20:' '
007FF6307C176D   C68424 9D000000 6A     mov byte ptr ss:[rsp+9D],6A         6A:'j'
007FF6307C1775   C68424 9E000000 6F     mov byte ptr ss:[rsp+9E],6F         6F:'o'
007FF6307C177D   C68424 9F000000 62     mov byte ptr ss:[rsp+9F],62         62:'b'
007FF6307C1785   C68424 A0000000 20     mov byte ptr ss:[rsp+A0],20         20:' '
007FF6307C178D   C68424 A1000000 3A     mov byte ptr ss:[rsp+A1],3A         3A:':'
007FF6307C1795   C68424 A2000000 29     mov byte ptr ss:[rsp+A2],29         29:')'
007FF6307C179D   C68424 A3000000 20     mov byte ptr ss:[rsp+A3],20         20:' '
007FF6307C17A5   C68424 A4000000 4D     mov byte ptr ss:[rsp+A4],4D         4D:'M'
007FF6307C17AD   C68424 A5000000 6F     mov byte ptr ss:[rsp+A5],6F         6F:'o'
007FF6307C17B5   C68424 A6000000 73     mov byte ptr ss:[rsp+A6],73         73:'s'
007FF6307C17BD   C68424 A7000000 74     mov byte ptr ss:[rsp+A7],74         74:'t'
007FF6307C17C5   C68424 A8000000 20     mov byte ptr ss:[rsp+A8],20         20:' '
007FF6307C17CD   C68424 A9000000 65     mov byte ptr ss:[rsp+A9],65         65:'e'
007FF6307C17D5   C68424 AA000000 78     mov byte ptr ss:[rsp+AA],78         78:'x'
007FF6307C17DD   C68424 AB000000 63     mov byte ptr ss:[rsp+AB],63         63:'c'
007FF6307C17E5   C68424 AC000000 65     mov byte ptr ss:[rsp+AC],65         65:'e'
007FF6307C17ED   C68424 AD000000 70     mov byte ptr ss:[rsp+AD],70         70:'p'
007FF6307C17F5   C68424 AE000000 74     mov byte ptr ss:[rsp+AE],74         74:'t'
007FF6307C17FD   C68424 AF000000 69     mov byte ptr ss:[rsp+AF],69         69:'i'
007FF6307C1805   C68424 B0000000 6F     mov byte ptr ss:[rsp+B0],6F         6F:'o'
007FF6307C180D   C68424 B1000000 6E     mov byte ptr ss:[rsp+B1],6E         6E:'n'
007FF6307C1815   C68424 B2000000 61     mov byte ptr ss:[rsp+B2],61         61:'a'
007FF6307C181D   C68424 B3000000 6C     mov byte ptr ss:[rsp+B3],6C         6C:'l'
007FF6307C1825   C68424 B4000000 21     mov byte ptr ss:[rsp+B4],21         21:'!'
007FF6307C182D   48:C74424 70 1D00000(  mov qword ptr ss:[rsp+70],1D
007FF6307C1836   41:B8 1D000000         mov r8d,1D
007FF6307C183C   48:8D9424 98000000     lea rdx,qword ptr ss:[rsp+98]
007FF6307C1844   48:8D4C24 78           lea rcx,qword ptr ss:[rsp+78]       [ss:[rsp+78]]:_register_onexit_function+A0
007FF6307C1849   E8 22060000            call ds-ctf-re1.7FF6307C1E70
007FF6307C184E   90                     nop
]]=[0000007B0C50FD88]=E0 'à'
```