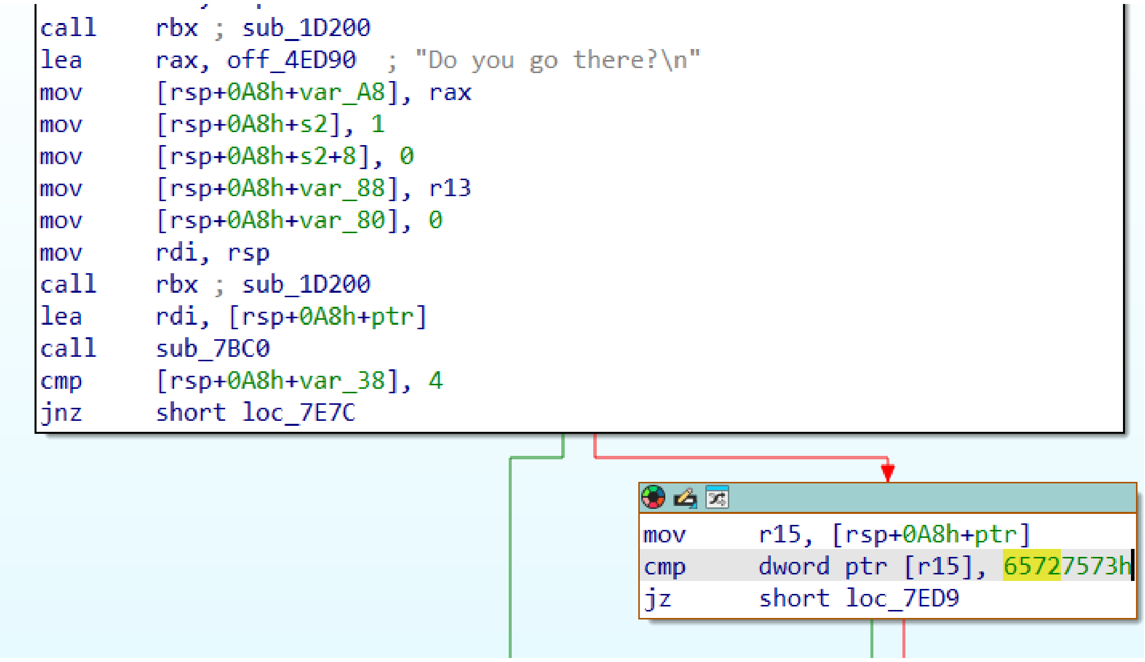
This is a 3-part chall.

Part 1: Do you go there?

To solve this part, we first need to localize the function which actually verifies thing. I used the strings view in ida to find the strings we needed, then pressed X to cross-reference. By doing this I got to the ida view needed.

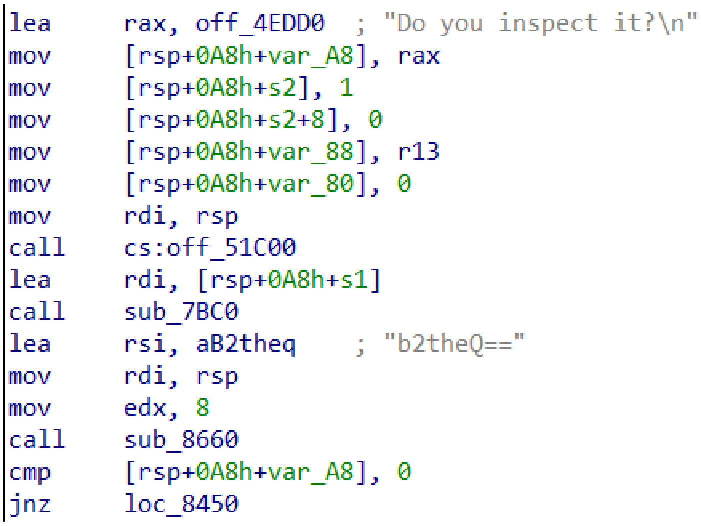


```
call    rbx ; sub_1D200
lea     rax, off_4ED90  ; "Do you go there?\n"
mov     [rsp+0A8h+var_A8], rax
mov     [rsp+0A8h+s2], 1
mov     [rsp+0A8h+s2+8], 0
mov     [rsp+0A8h+var_88], r13
mov     [rsp+0A8h+var_80], 0
mov     rdi, rsp
call    rbx ; sub_1D200
lea     rdi, [rsp+0A8h+ptr]
call    sub_7BC0
cmp     [rsp+0A8h+var_38], 4
jnz     short loc_7E7C
```

```
mov     r15, [rsp+0A8h+ptr]
cmp     dword ptr [r15], 65727573h
jz      short loc_7ED9
```

From my understanding, the input is verified to have 4 chars and to correspond to the hex 65727573 witch decodes to “sure”. Sending “sure” is the correct response, so we got our first answer.

Part 2: Do you inspect it?

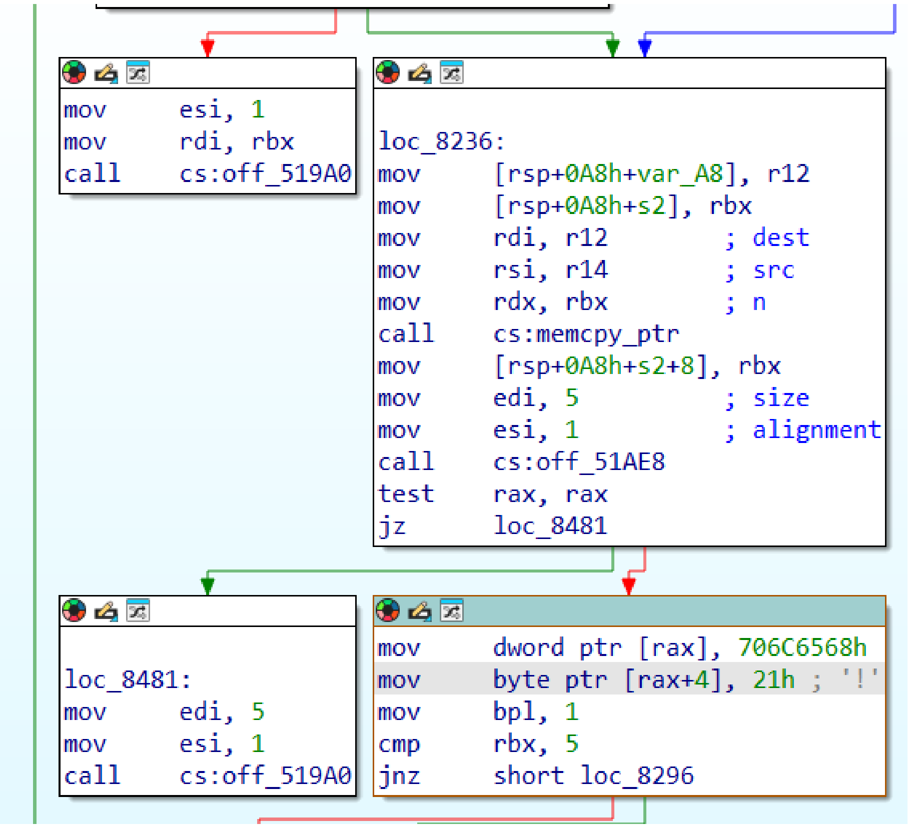


```
lea     rax, off_4EDD0  ; "Do you inspect it?\n"
mov     [rsp+0A8h+var_A8], rax
mov     [rsp+0A8h+s2], 1
mov     [rsp+0A8h+s2+8], 0
mov     [rsp+0A8h+var_88], r13
mov     [rsp+0A8h+var_80], 0
mov     rdi, rsp
call    cs:off_51C00
lea     rdi, [rsp+0A8h+s1]
call    sub_7BC0
lea     rsi, aB2theq    ; "b2theQ=="
mov     rdi, rsp
mov     edx, 8
call    sub_8660
cmp     [rsp+0A8h+var_A8], 0
jnz     loc_8450
```

Looking further down in the ida view discovered in part 1 reveals that the 2nd answer corresponds to the base64 strings b2theQ==, which decodes to “okay”. Sending “okay” is the correct response, so we got our second answer.

Part 3: What do you say to it?

Finding the answer to this a bit more tricky, but we just need to follow the same pattern. Looking through the ida view we stumble upon this:



```
mov     esi, 1
mov     rdi, rbx
call    cs:off_519A0
```

```
loc_8236:
mov     [rsp+0A8h+var_A8], r12
mov     [rsp+0A8h+s2], rbx
mov     rdi, r12        ; dest
mov     rsi, r14        ; src
mov     rdx, rbx        ; n
call    cs:memcpy_ptr
mov     [rsp+0A8h+s2+8], rbx
mov     edi, 5          ; size
mov     esi, 1          ; alignment
call    cs:off_51AE8
test    rax, rax
jz      loc_8481
```

```
loc_8481:
mov     edi, 5
mov     esi, 1
call    cs:off_519A0
```

```
mov     dword ptr [rax], 706C6568h
mov     byte ptr [rax+4], 21h ; '!'
mov     bpl, 1
cmp     rbx, 5
jnz     short loc_8296
```

I’m unsure how the verification occurs and what happens, but I see another hex string, 706c6568 corresponding to “help” and a “!” at the end. As a result, I try to send “help!” and it is the correct result, thus finishing the challenge.

Made with love by: AndreiCat