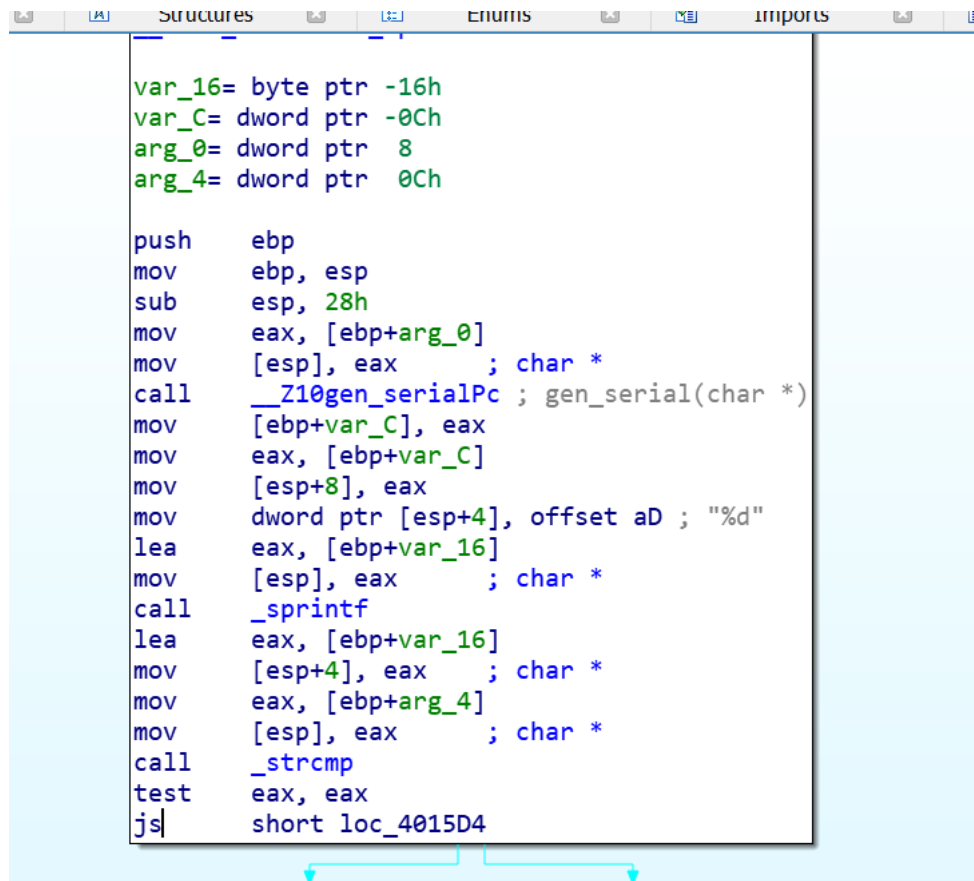


Как делать РК?

1. Устанавливаем IDA https://www.hex-rays.com/products/ida/support/download_freeware.shtml и Windows

2. Открываем файл.exe в IDA



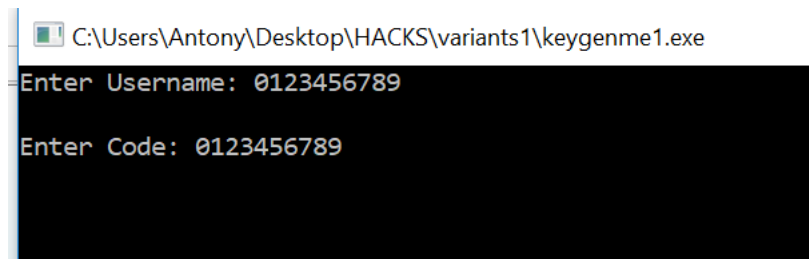
The screenshot shows the IDA Pro interface with the 'Structures' tab selected. The assembly view displays the following code:

```
var_16= byte ptr -16h
var_C= dword ptr -0Ch
arg_0= dword ptr 8
arg_4= dword ptr 0Ch

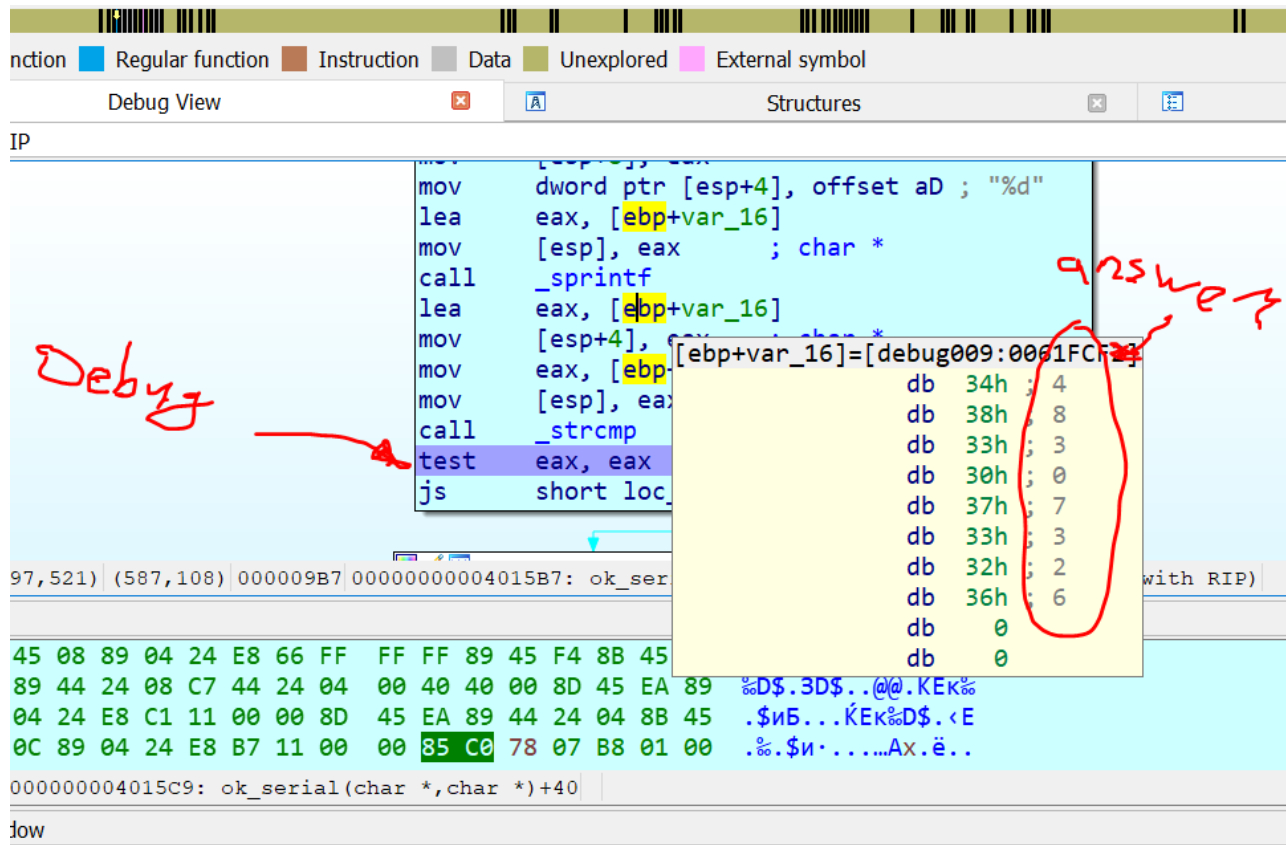
push    ebp
mov     ebp, esp
sub     esp, 28h
mov     eax, [ebp+arg_0]
mov     [esp], eax ; char *
call    __Z10gen_serialPc ; gen_serial(char *)
mov     [ebp+var_C], eax
mov     eax, [ebp+var_C]
mov     [esp+8], eax
mov     dword ptr [esp+4], offset aD ; "%d"
lea     eax, [ebp+var_16]
mov     [esp], eax ; char *
call    _sprintf
lea     eax, [ebp+var_16]
mov     [esp+4], eax ; char *
mov     eax, [ebp+arg_4]
mov     [esp], eax ; char *
call    _strcmp
test    eax, eax
js      short loc_4015D4
```

Это код в ассемблере, можно ставить брейкпоинты -> запускать и смотреть что в регистрах

3. Запускаем прогу в режиме дебага

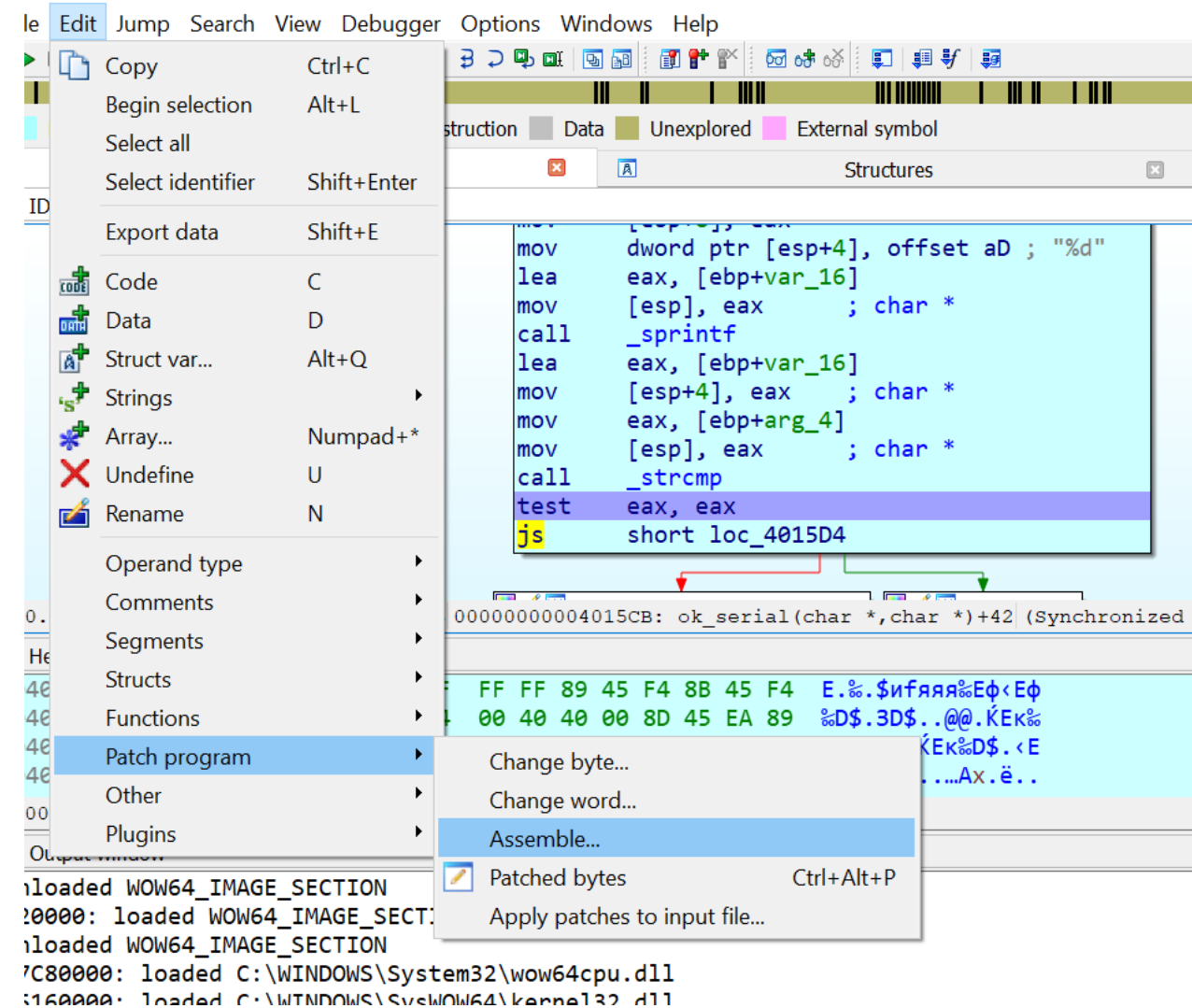


Вводим случайные данные



Палим настоящий пароль, т.к. там банально сравнивается `your_pass == true_pass`
Мы "подглядываем" `true_pass`

4. Делаем патч программы -> что бы вне зависимости от пароля мы проходили по условию, это функция "jnz" после test eax, eax меняем ее на "js". На скрине уже сделан патч.



5. Мы узнали конкретную пару валидных данных(0123456789, 48307326), напишем keygen на питоне

Будем палить assembler и пытаться перевести в python

```

; Attributes: bp-based frame

; _DWORD __cdecl gen_serial(char *)
public __Z10gen_serialPc
__Z10gen_serialPc proc near

var_1A      = byte ptr -1Ah
var_19      = byte ptr -19h
var_18      = dword ptr -18h
var_14      = dword ptr -14h
var_10      = dword ptr -10h
var_A       = byte ptr -0Ah
var_9       = byte ptr -9
var_8       = dword ptr -8
var_4       = dword ptr -4
arg_0       = dword ptr 8

push        ebp
mov         ebp, esp
sub         esp, 20h
mov         [ebp+var_A], 0Ch
mov         [ebp+var_4], 455FADh
mov         [ebp+var_10], 2A47D56h
mov         [ebp+var_14], 13h
mov         [ebp+var_18], 88h
mov         eax, [ebp+arg_0]
mov         [ebp+var_8], eax
mov         [ebp+var_19], 0
jmp         short loc_401574

01500: gen_serial(char *) (Synchronized with Hex View-1)
```

<https://www.codeproject.com/Articles/15971/Using-Inline-Assembly-in-C-C>