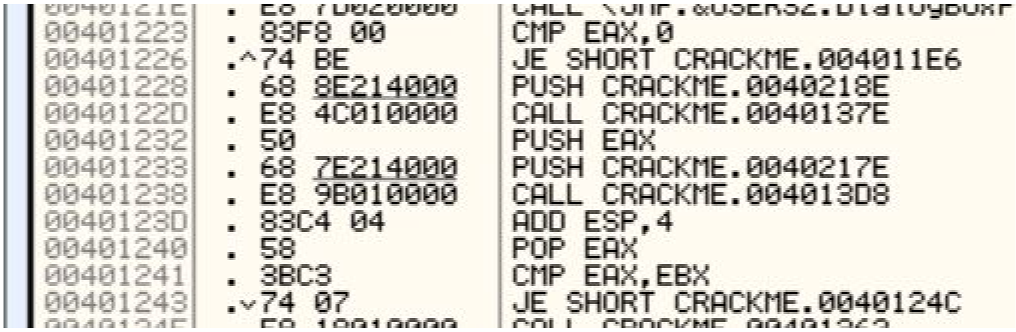
Elizabeth Ye
ely5

HW3 Part 1

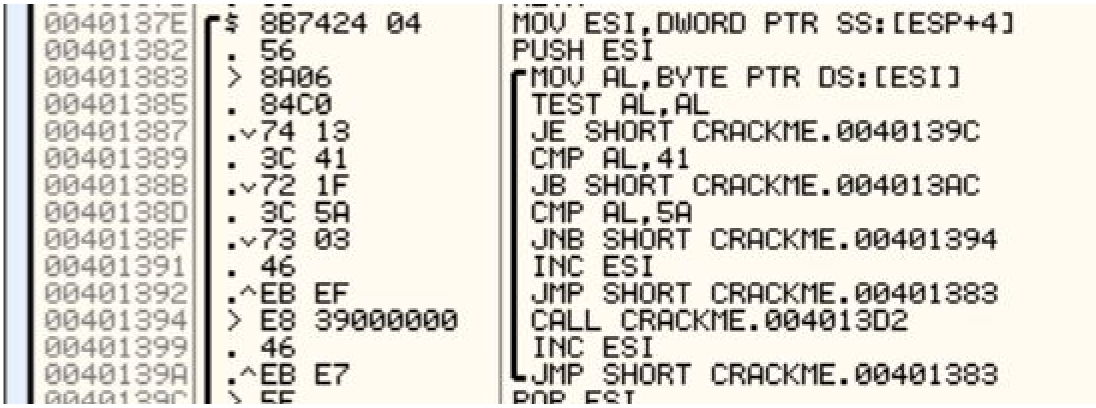
Problem 2

I used the OllyDbg tools to set/remove breakpoints, step through/over, and run/pause. Using these tools helped me find the following sections that test and calculate input and output. From the previous problem I saw that the main driver code was at 00401223



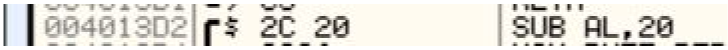
```
00401223  . 83F8 00       CMP EAX,0
00401226  .^74 BE         JE SHORT CRACKME.004011E6
00401228  . 68 8E214000   PUSH CRACKME.0040218E
0040122D  . E8 4C010000   CALL CRACKME.0040137E
00401232  . 50            PUSH EAX
00401233  . 68 7E214000   PUSH CRACKME.0040217E
00401238  . E8 9B010000   CALL CRACKME.004013D8
0040123D  . 83C4 04       ADD ESP,4
00401240  . 58            POP EAX
00401241  . 3BC3          CMP EAX,EBX
00401243  .v74 07         JE SHORT CRACKME.0040124C
```

CMP EAX,EBX is checking to see whether the user inputted the calculated serial.



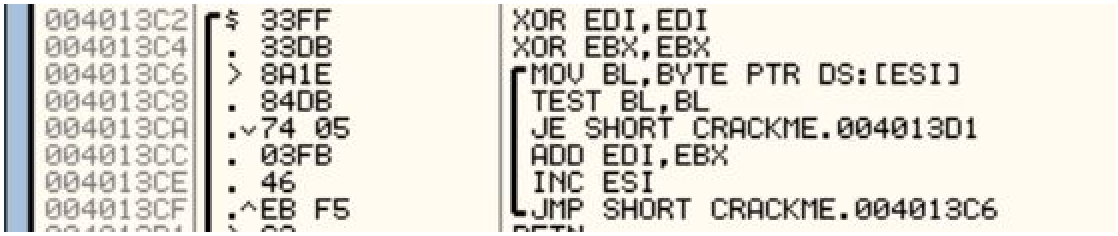
```
0040137E  $ 8B7424 04     MOV ESI,DWORD PTR SS:[ESP+4]
00401382  . 56            PUSH ESI
00401383  > 8A06          MOV AL,BYTE PTR DS:[ESI]
00401385  . 84C0          TEST AL,AL
00401387  .v74 13         JE SHORT CRACKME.0040139C
00401389  . 3C 41         CMP AL,41
0040138B  .v72 1F         JB SHORT CRACKME.004013AC
0040138D  . 3C 5A         CMP AL,5A
0040138F  .v73 03         JNB SHORT CRACKME.00401394
00401391  . 46            INC ESI
00401392  .^EB EF         JMP SHORT CRACKME.00401383
00401394  > E8 39000000   CALL CRACKME.004013D2
00401399  . 46            INC ESI
0040139A  .^EB E7         JMP SHORT CRACKME.00401383
```

This section checks whether this input name is valid, all letters between A and Z. The call at 00401394 jumps to 004013D2



```
004013D2  $ 2C 20         SUB AL,20
```

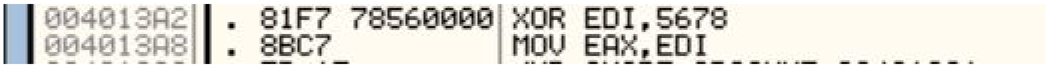
This converts all lowercase letters to uppercase letters.



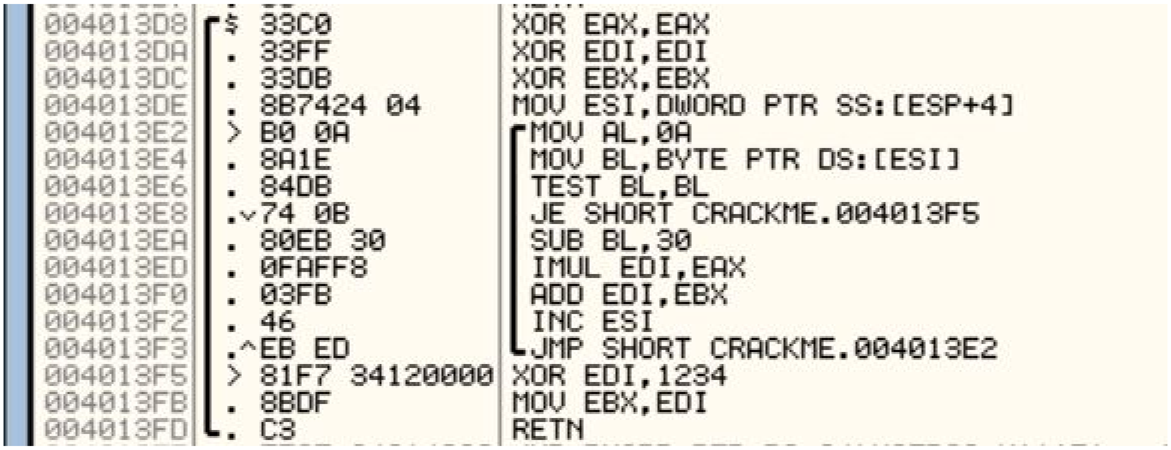
```
004013C2  $ 33FF        XOR EDI,EDI
004013C4  . 33DB        XOR EBX,EBX
004013C6  > 8A1E        MOV BL,BYTE PTR DS:[ESI]
004013C8  . 84DB        TEST BL,BL
004013CA  .v74 05       JE SHORT CRACKME.004013D1
004013CC  . 03FB        ADD EDI,EBX
004013CE  . 46          INC ESI
004013CF  .^EB F5       JMP SHORT CRACKME.004013C6
```

This is the code that creates the serial number for each valid name entered. It loops through and sums the hex values of each character, creating the serial number.

For my name: LIZ = 4C + 49 + 5A = EF
Then this value is XOR-d with 5678 as shown below.
EF XOR 5678 = 5697



```
004013A2  . 81F7 78560000 XOR EDI,5678
004013A8  . 8BC7          MOV EAX,EDI
```

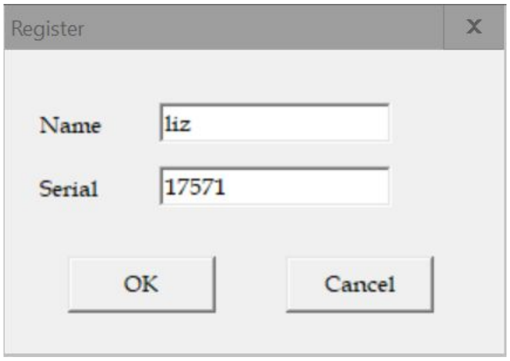


```
004013D8  $ 33C0          XOR EAX,EAX
004013DA  . 33FF          XOR EDI,EDI
004013DC  . 33DB          XOR EBX,EBX
004013DE  . 8B7424 04     MOV ESI,DWORD PTR SS:[ESP+4]
004013E2  > B0 0A         MOV AL,0A
004013E4  . 8A1E          MOV BL,BYTE PTR DS:[ESI]
004013E6  . 84DB          TEST BL,BL
004013E8  .v74 0B         JE SHORT CRACKME.004013F5
004013EA  . 80EB 30       SUB BL,30
004013ED  . 0FAFF8        IMUL EDI,EAX
004013F0  . 03FB          ADD EDI,EBX
004013F2  . 46            INC ESI
004013F3  .^EB ED         JMP SHORT CRACKME.004013E2
004013F5  > 81F7 34120000 XOR EDI,1234
004013FB  . 8BDF          MOV EBX,EDI
004013FD  . C3            RETN
```

This call to 004013D8 is the part that converts the entered serial number (in decimal) to hexadecimal. This result, XOR-d with 1234 should equal 5697, the calculated value. We can reverse engineer this using the properties of XOR to find the solution:

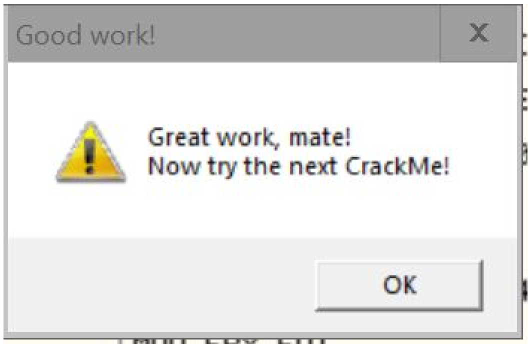
5697 XOR 1234 = 44A3 -> 17571

This worked!



Name liz

Serial 17571



Great work, mate!
Now try the next CrackMe!

This method can be used for any name, to find its corresponding serial number.