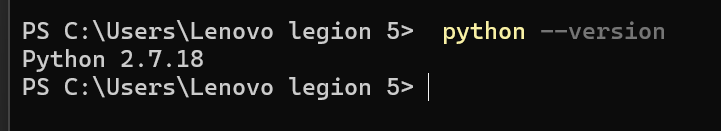
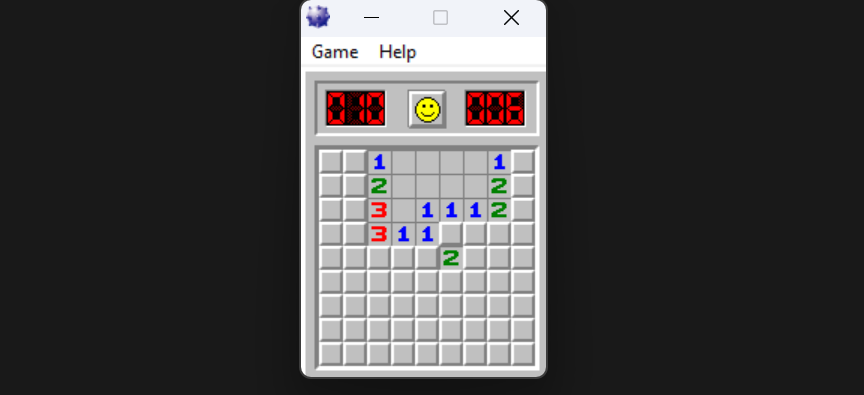
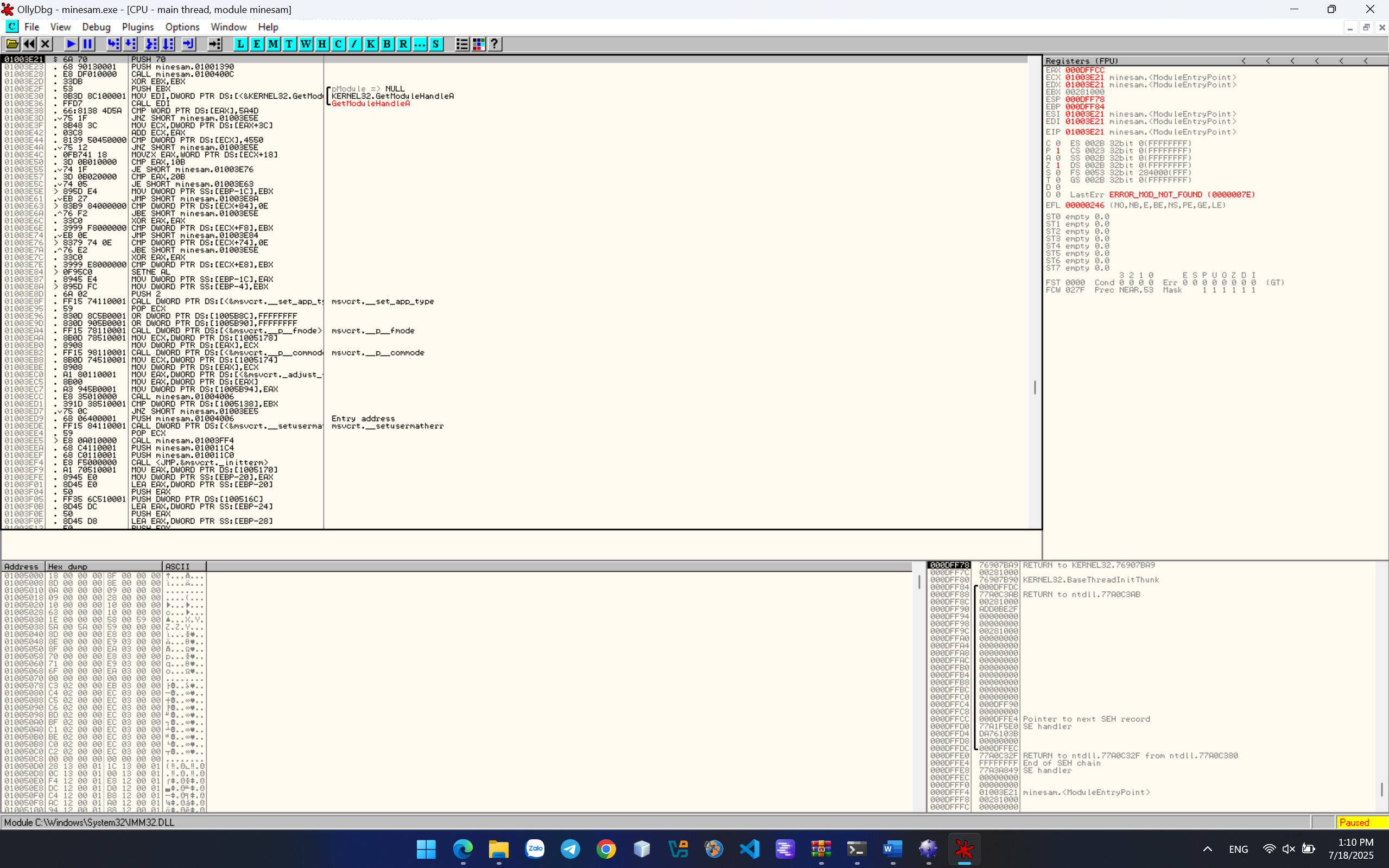
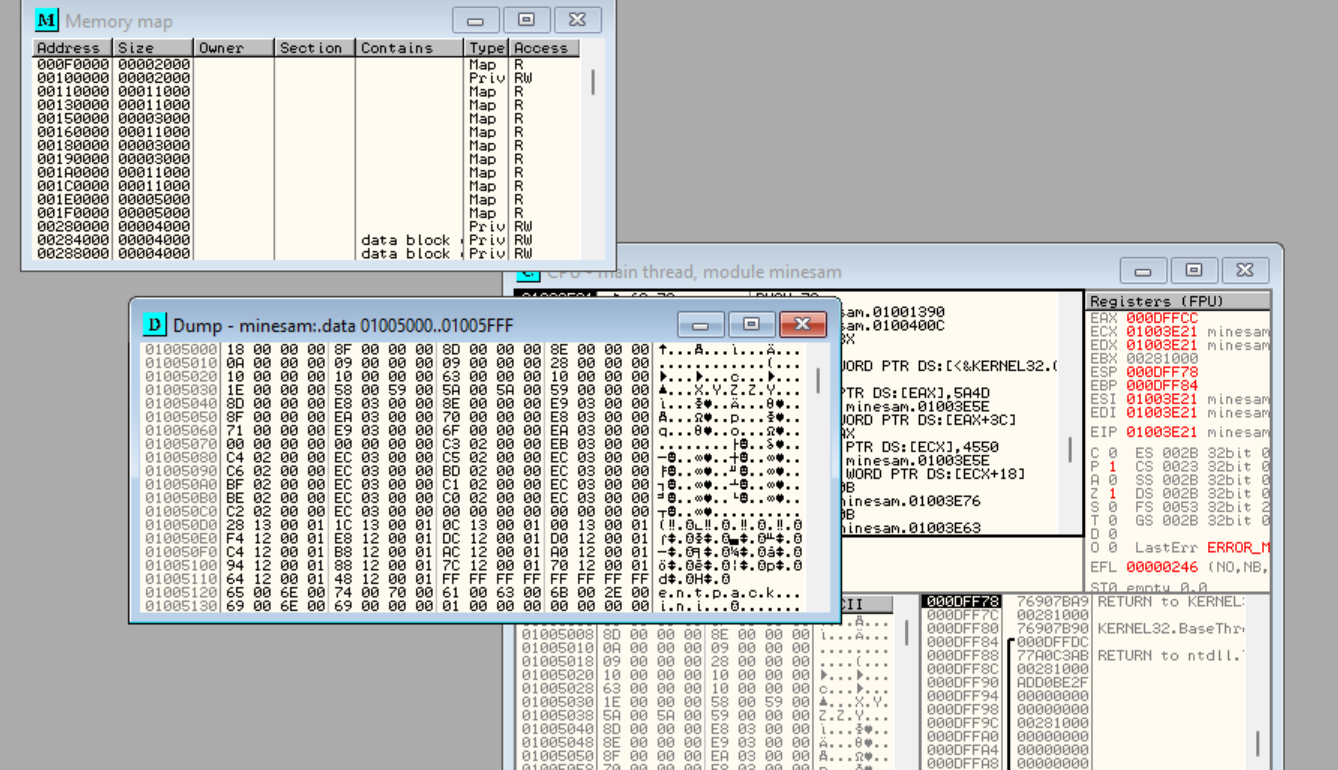
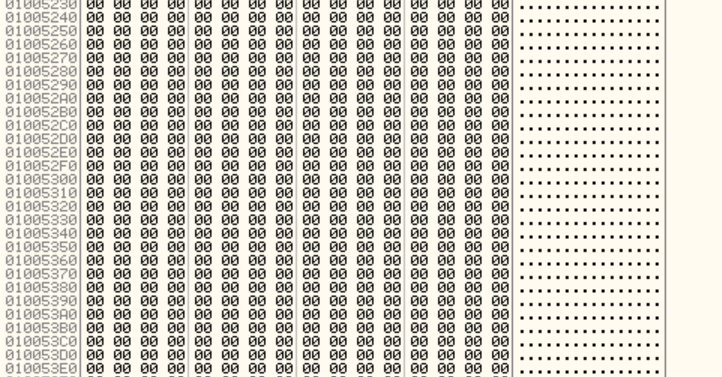
**Lab 18.1: Hacking Minesweeper with Ollydbg**

python --version

Getting Minesweeper

Viewing the Game in OllyDbg

From the OllyDbg menu bar, click View, Memory. The memory segments are shown, as shown below. Right-click the minesam .data line and click Dump, as shown below. 

In the Dump window, scroll down to show memory near 01005340. This area contains only zeroes, as shown below

o From the OllyDbg menu bar, click View, CPU.

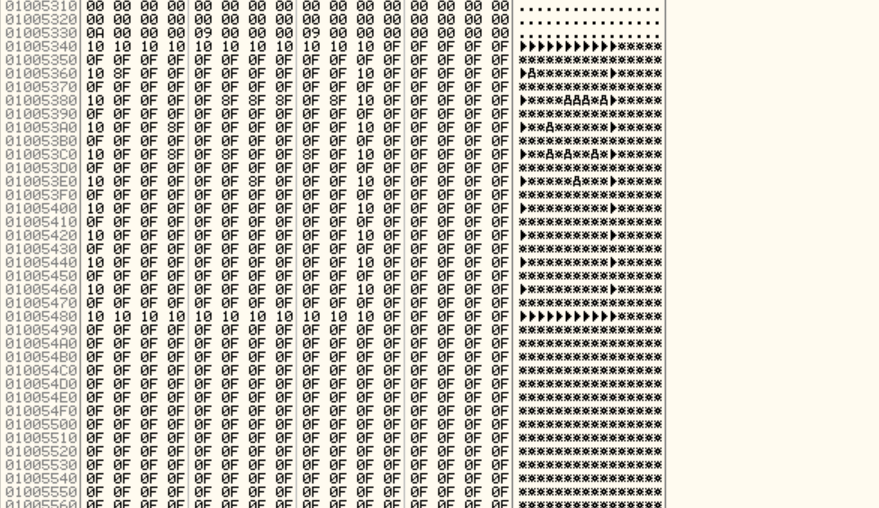
o From the OllyDbg menu bar, click Debug, Run.

o A Minesweeper window opens, but does not come to the front. Click its button on the taskbar to bring it to the front, as shown below

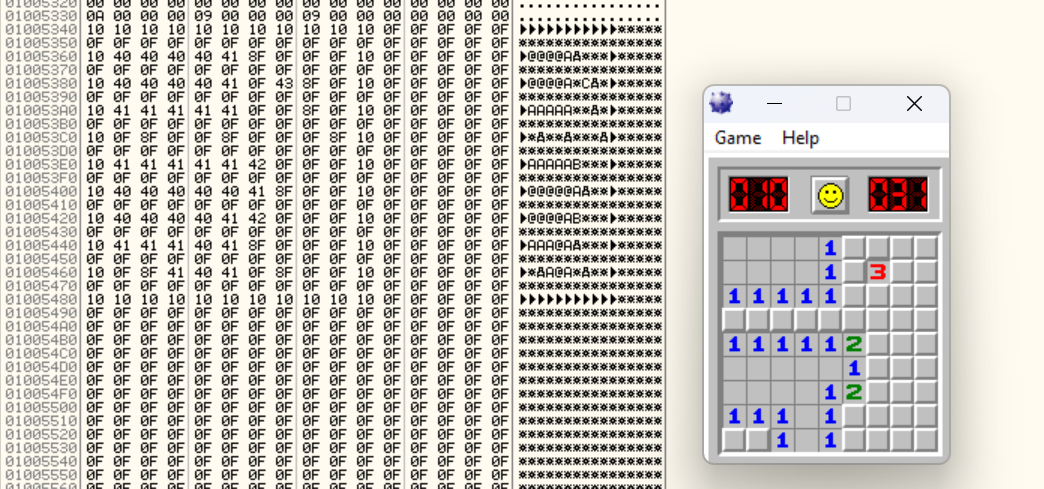
**• Viewing the Stored Gameboard**

o From the OllyDbg menu bar, click Window, Dump.

o The memory after 01005340 now contains data, as shown below



o Click the Minesweeper button on the taskbar to bring it to the front.

o Compare the Minesweeper gameboard with the Dump window. You can see that the gameboard is stored in RAM, using an "A" for "1", and a "B" for "2", as shown below. 

o If we can read the RAM, we can cheat at the game.

o Notice the green-highlighted region in the image above. If we can find this sequence of bytes in RAM, we can find the gameboard in a memory dump

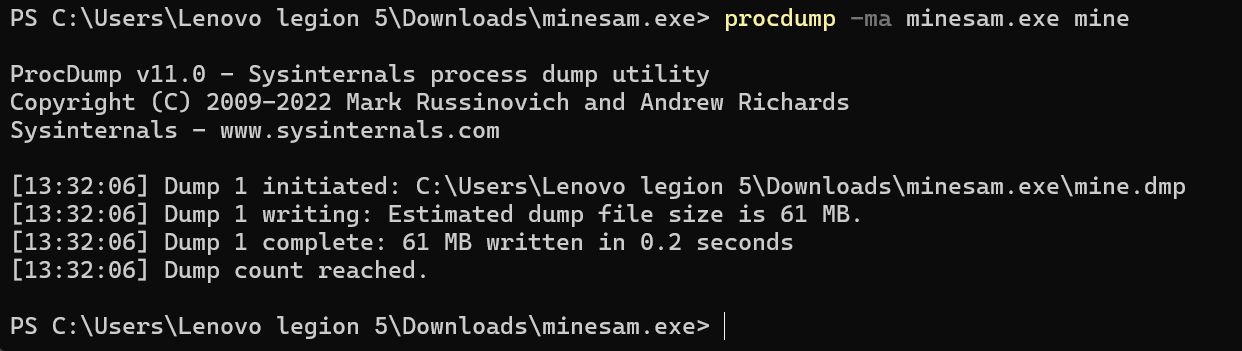
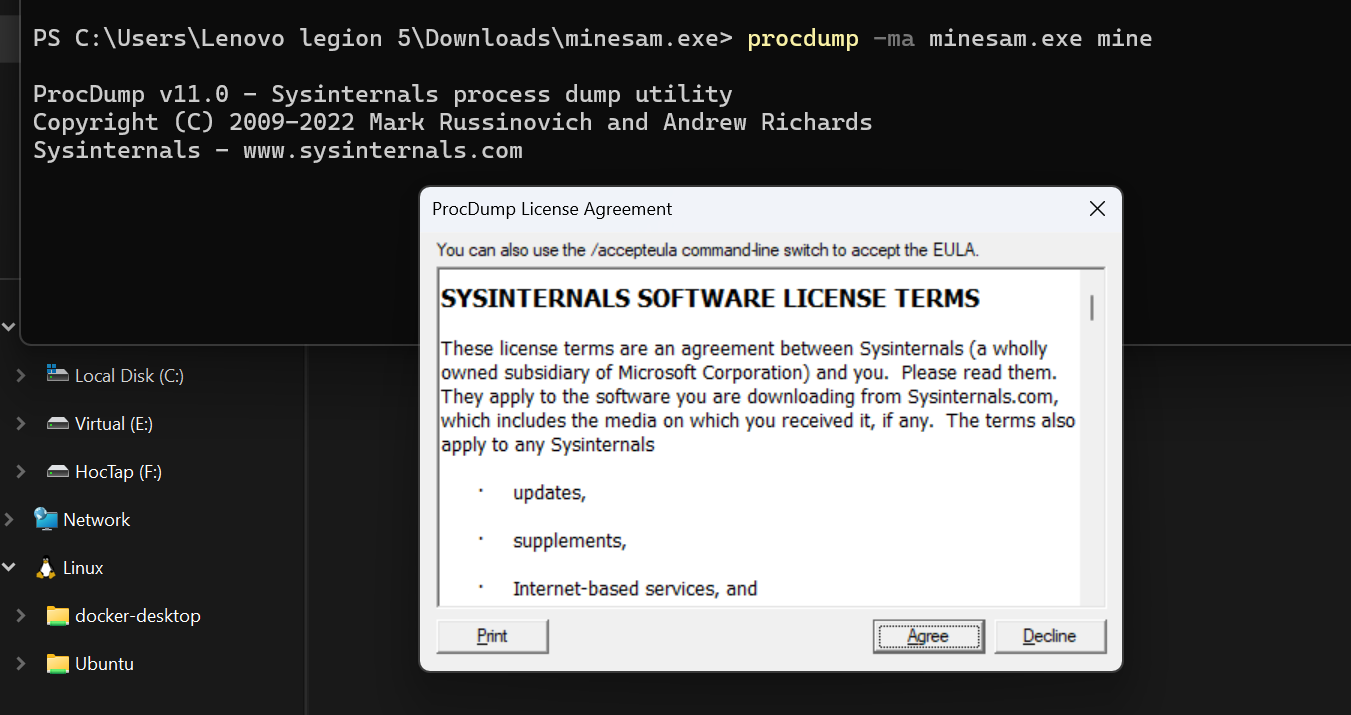
**• Getting Procdump**

**• Capturing Process Memory**

o Close Minesweeper. Close OllyDbg. Double-click minesam.exe to run

o Open a Command Prompt and execute these commands: cd C:\Users\Administrator\Deskstop\ procdump -ma minesam.exe mine

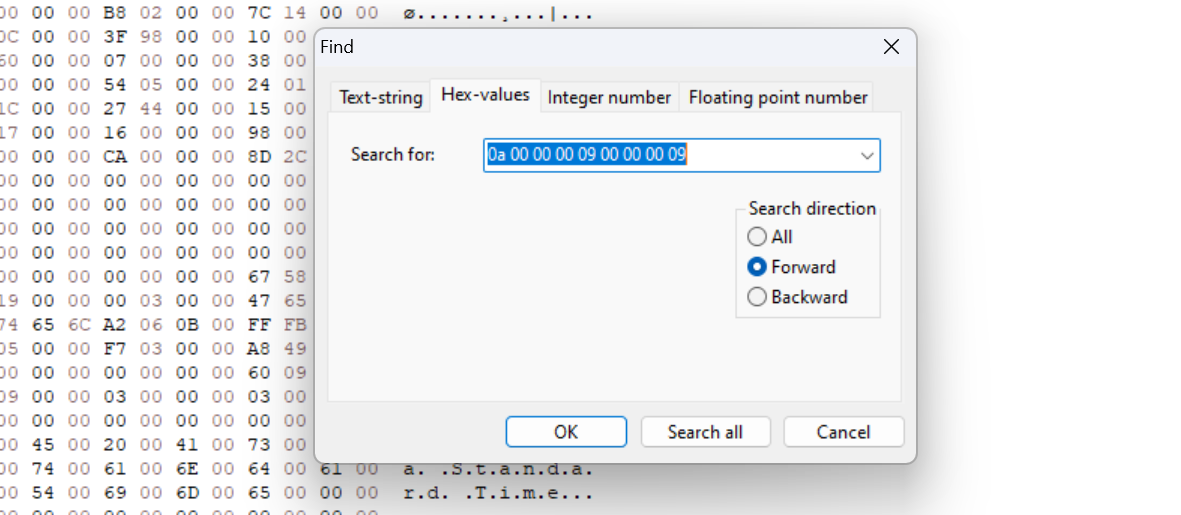
o A box pops up, titled ProcDump License Agreement. Click Agree.

o Procdump makes a dump file, as shown below. 

**• Viewing the Memory with HxD**

o From the HxD menu bar, click Search, Find.

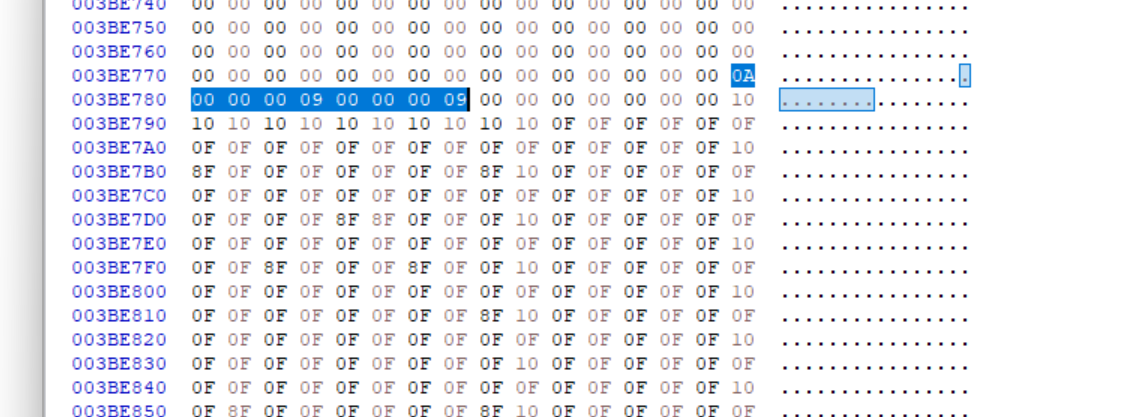
o In the "Find" field, select a Datatype of Hex-values.

o In the "Search for" field, enter this text, as shown below **0a 00 00 00 09 00 00 00 09**

o In the "Find" box, click OK.

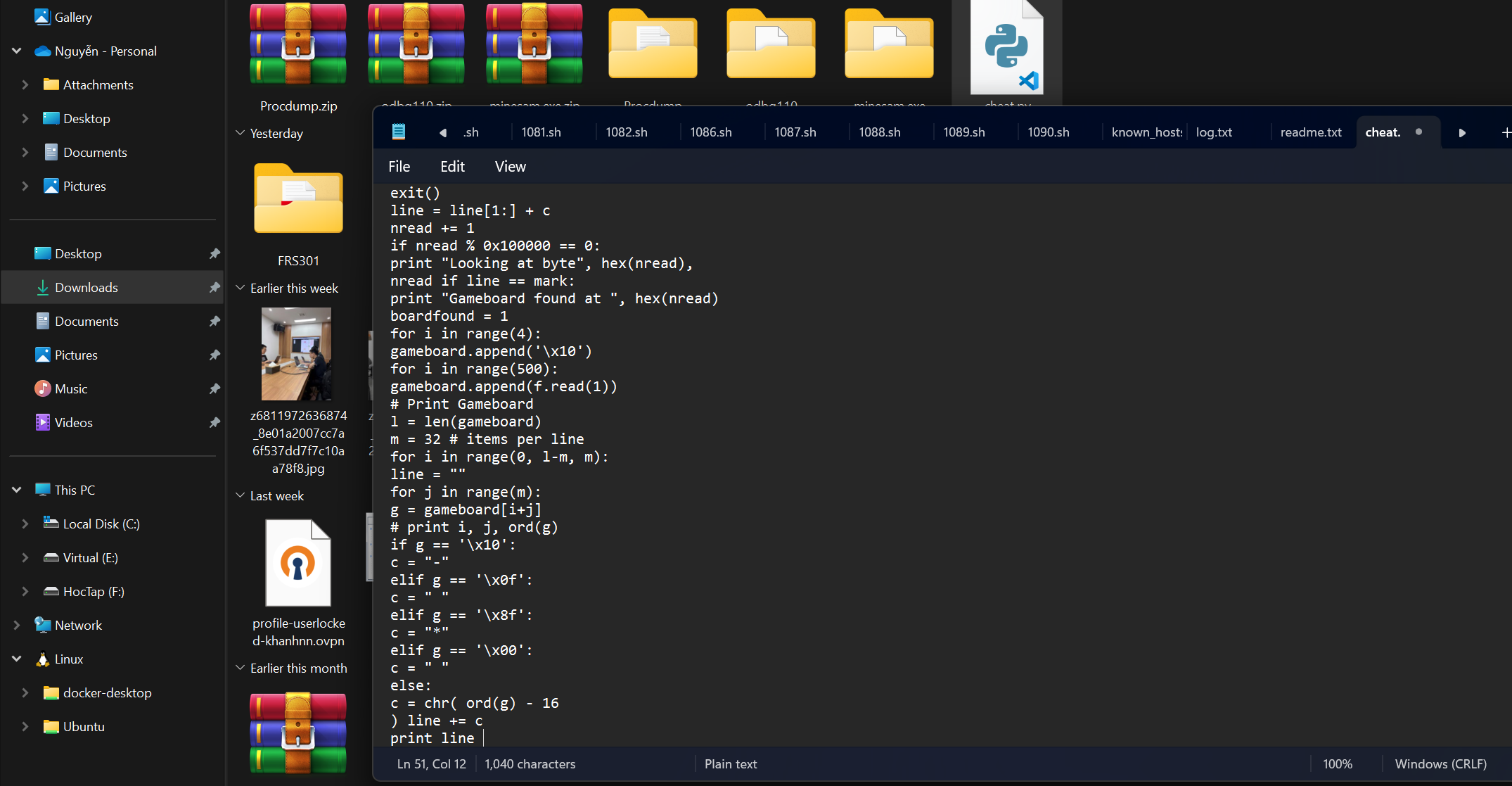
o The string is found, but it may not be the correct hit. The first one doesn't have the gameboard after it, as shown below

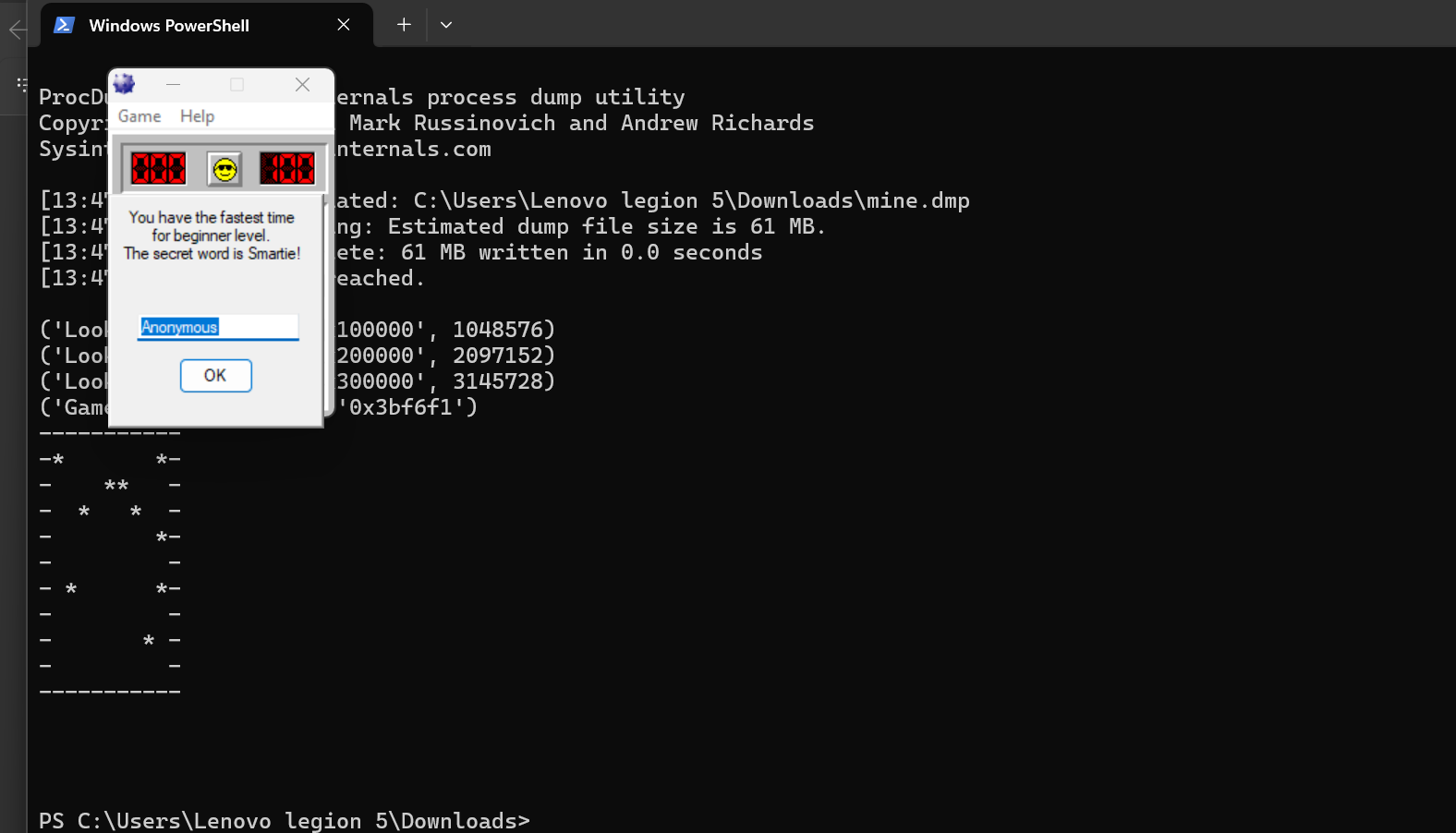
o From the HxD menu bar, click Search, "Find again".

o This time it finds the gameboard data, as shown below

**• Creating a Python Script**

o In the Command Prompt window, execute this command: python cheat.py

o The program shows the location of the mines. With this information, you should easily be able to click all the squares without mines, as shown below. 

o When you win the game, a secret word will appear, which is covered by a green box in the image below. 

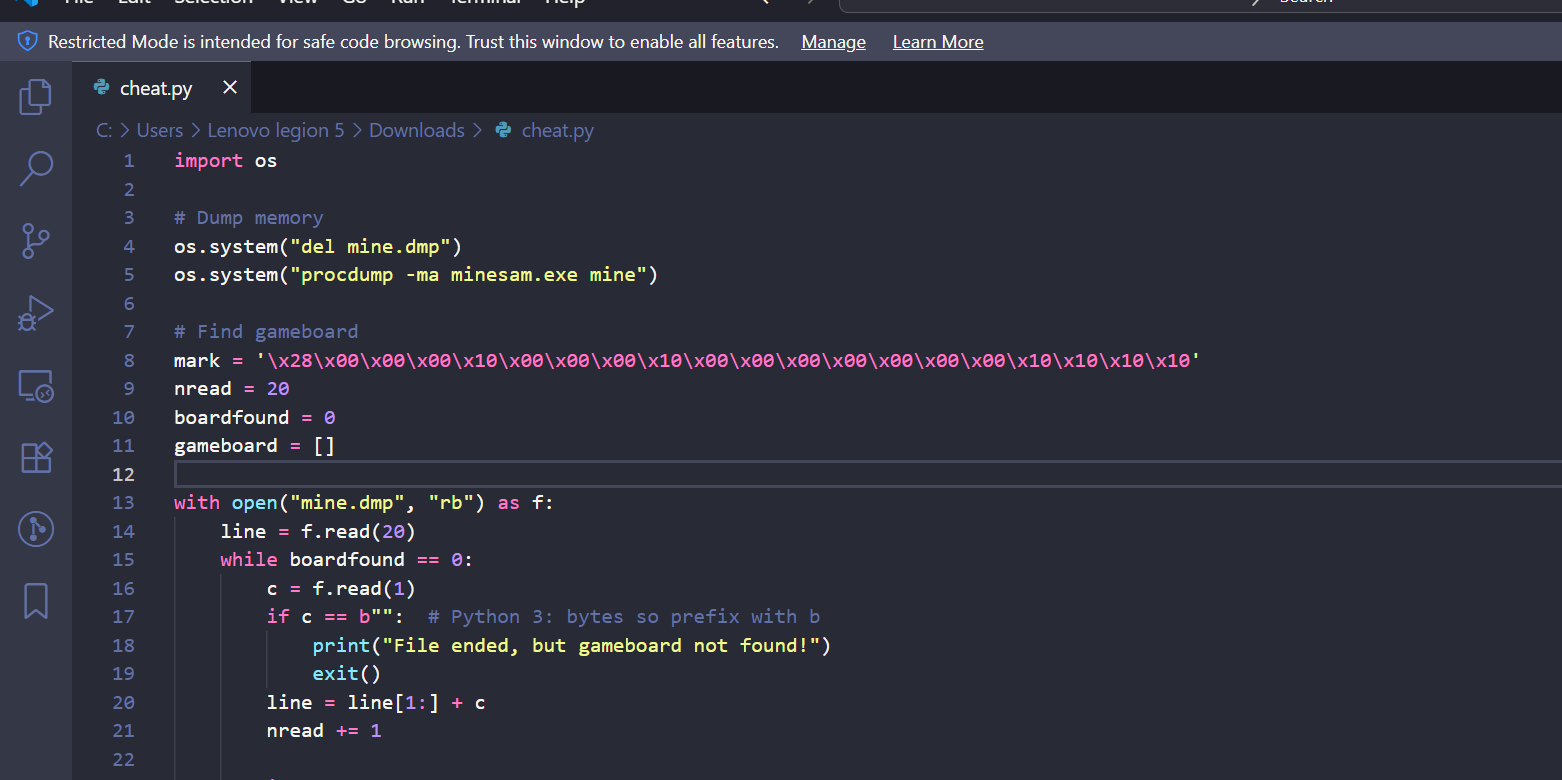
**• Intermediate Level**

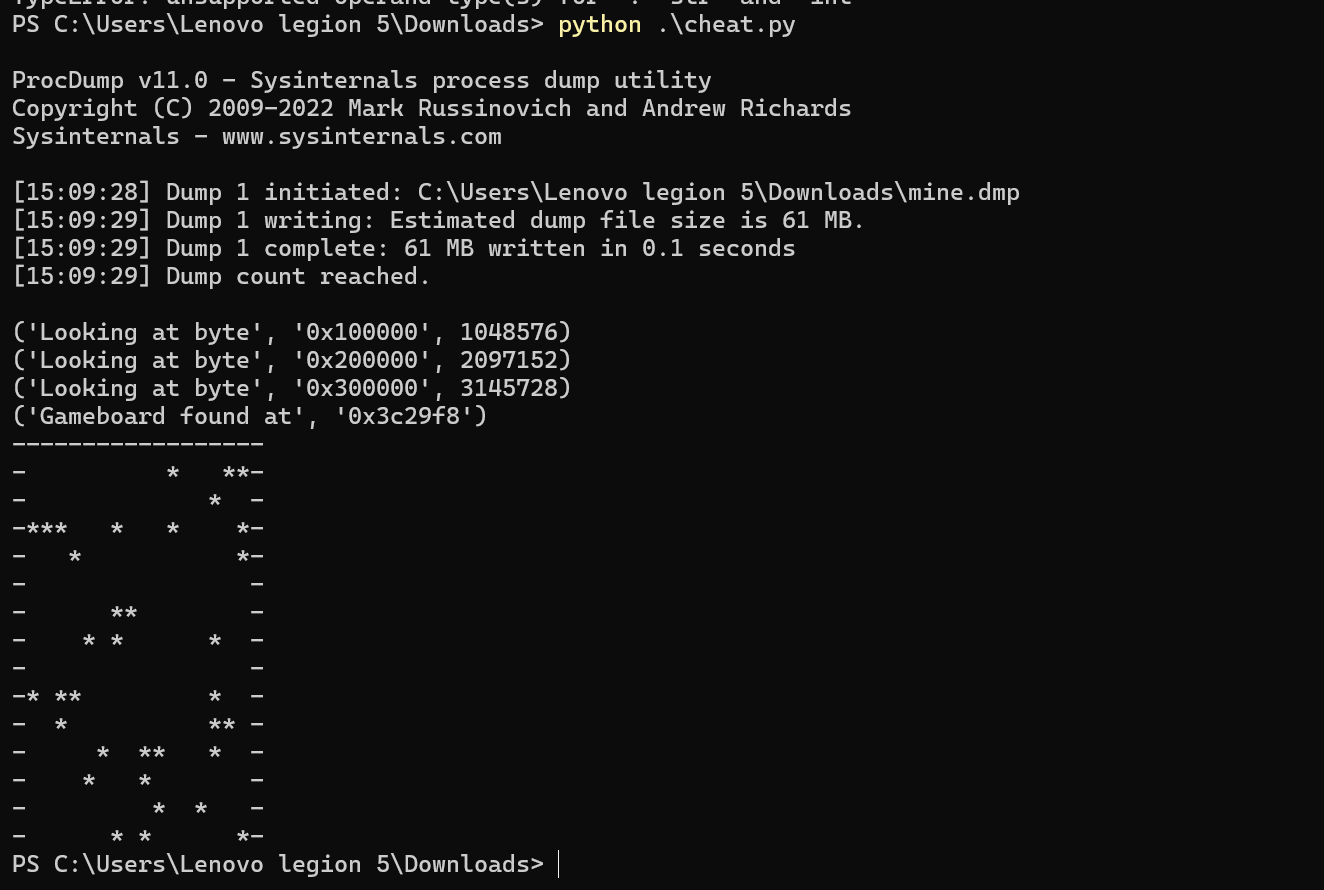
o In Minesweeper, click Game, Intermediate.

o Create a cheating tool that works for this level and win the game, as shown below.

o Hint: Search for 10 10 10 10 to find the gameboard

o Open a Command Prompt and execute these commands: procdump -ma minesam.exe mine

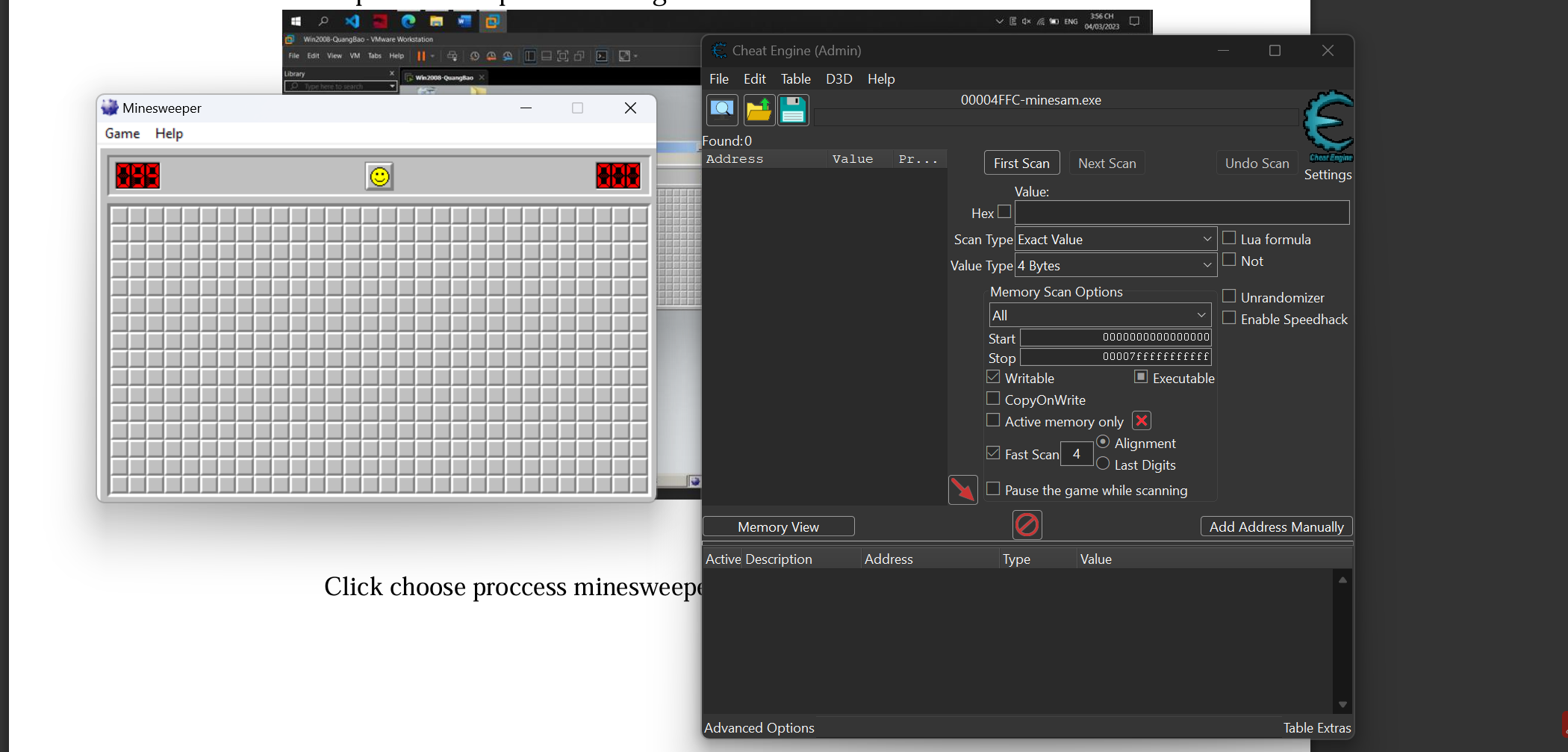
o Replace “\x28\x00\x00\x00\x10\x00\x00\x00\x10\x00\x00\x00\x00\x00\x00\x00\x10\x10\x10\x10” in mark on script

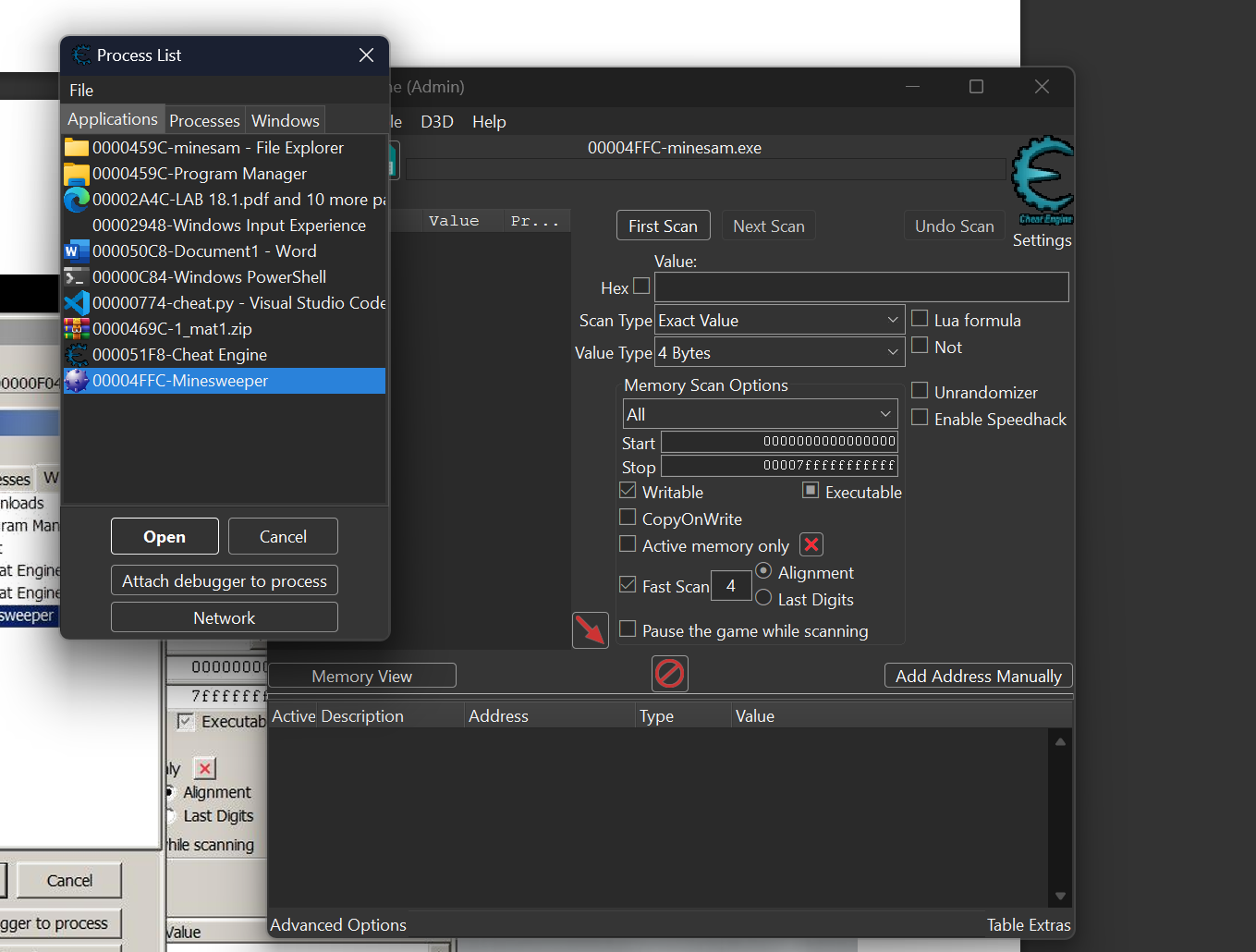
o Run the script 

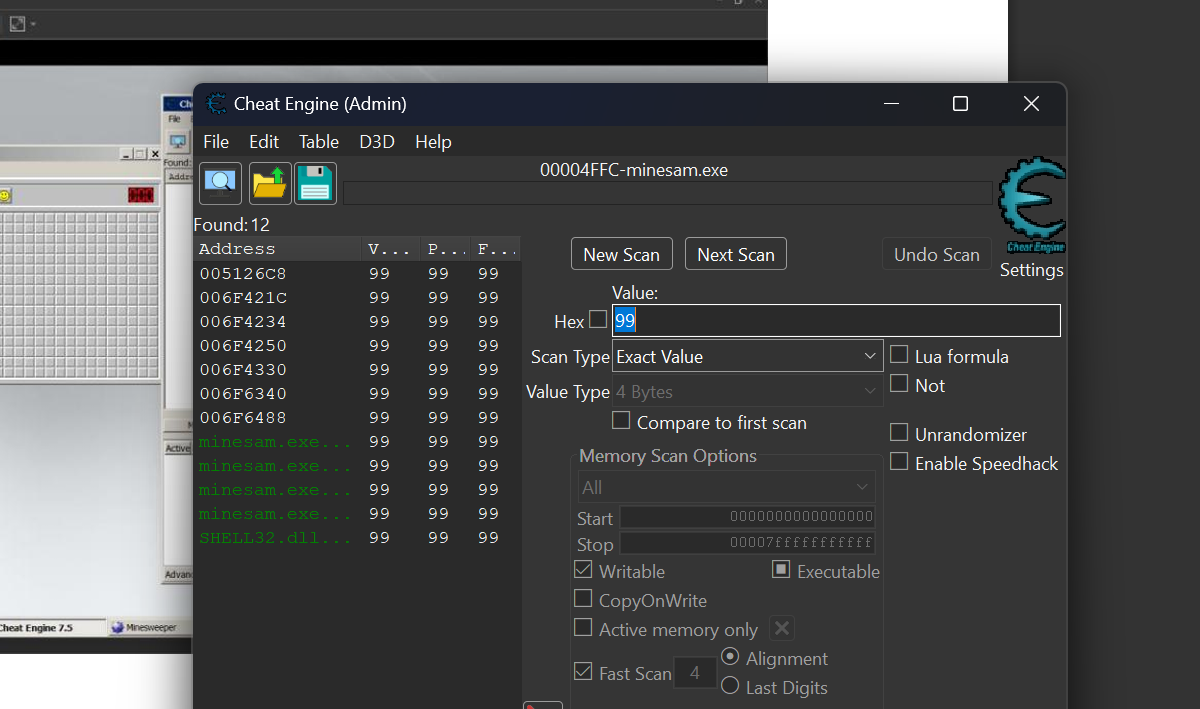
**Expert Level**

o Using Cheat Engine\

o Change the number of boom in minesweeper

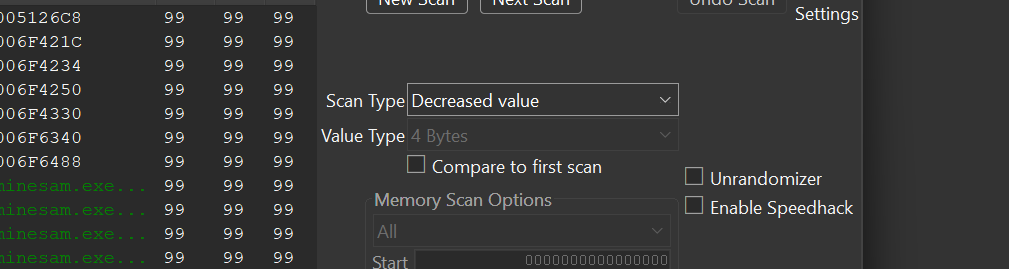
First open minesweeper in Expert Level Open Cheat Engine

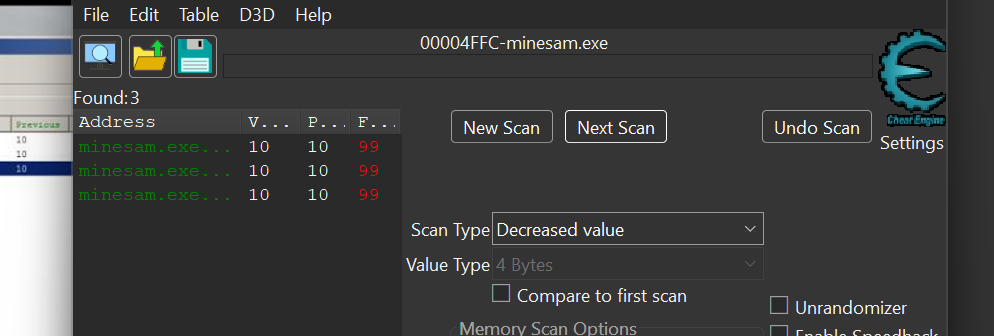
Click choose proccess minesweeper

Type 99 values and press Enter - 99 is the number of the booms

Click first scan

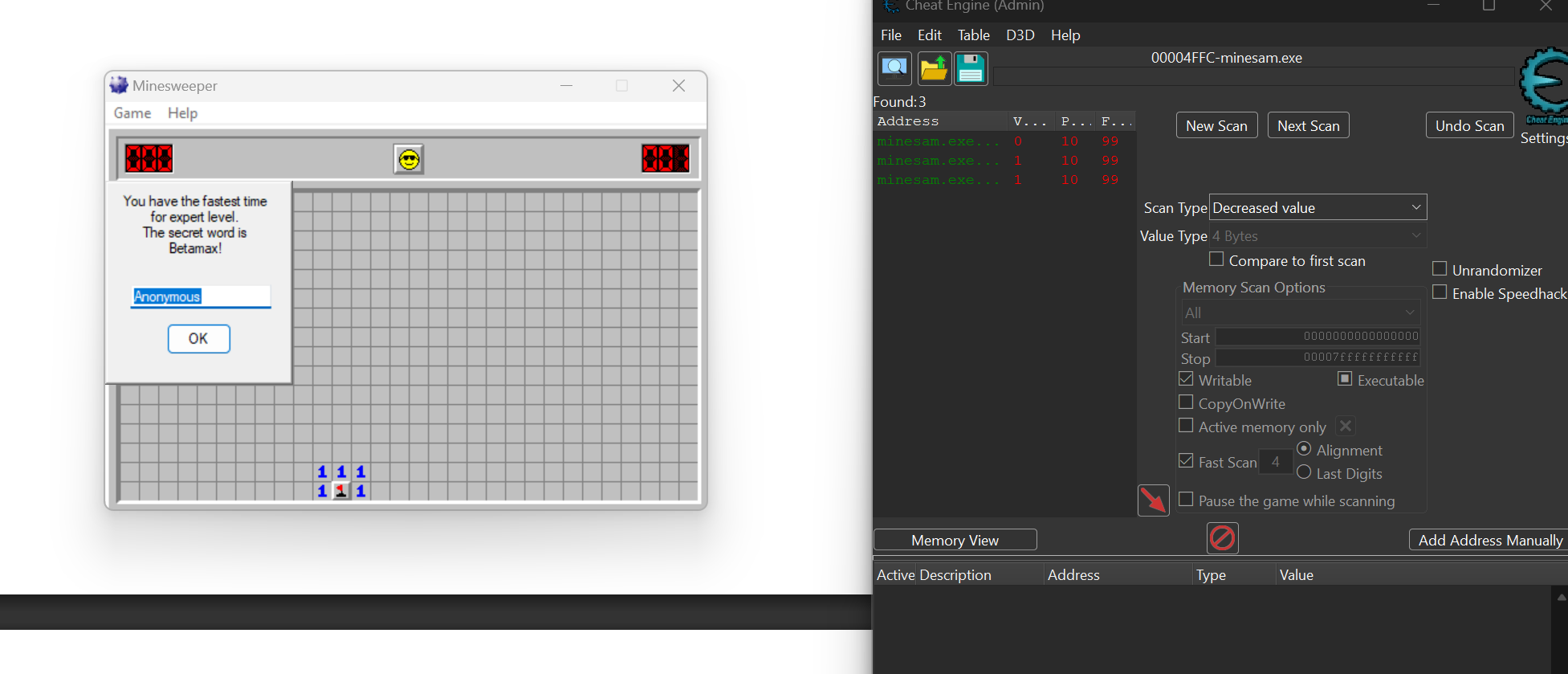
After first scan, click to Decreased Value in Scan type

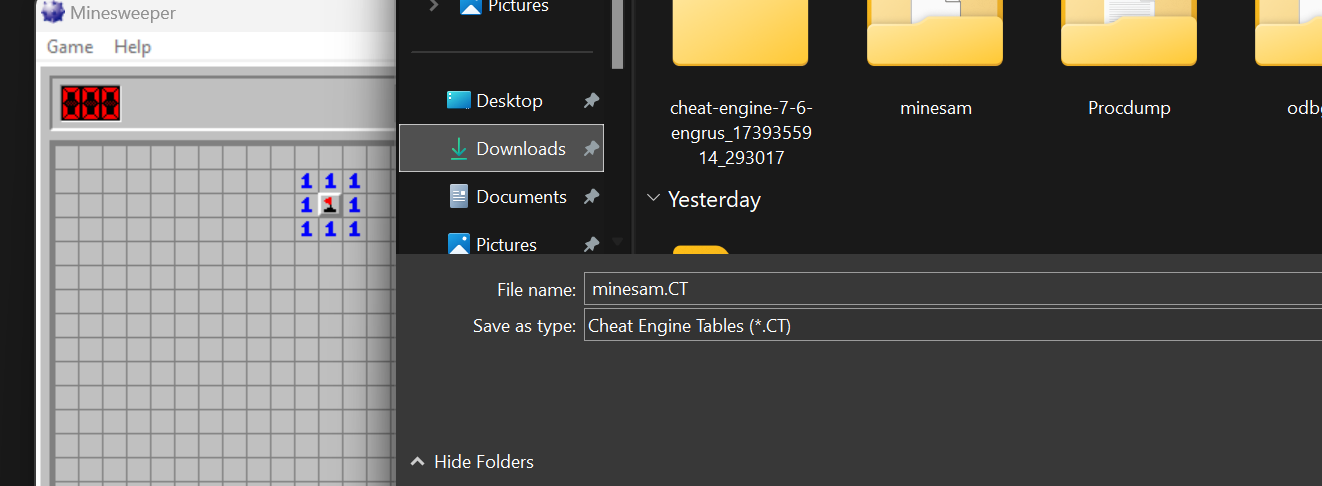
Change game mode into Beginer

Click next scan

Choose all results and hit “Ctrl + E” to edit value

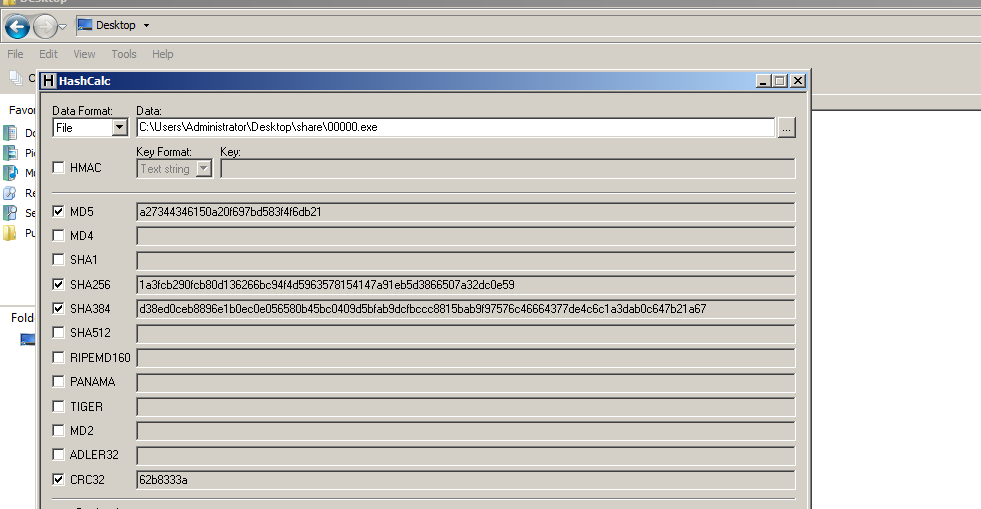
Change all to “1”, Click Enter

Now change the game mode to Expert level and click random box

You won the game! Right Click and chose Add to new group Change name to Minesweeper And pess “Ctrl + s” to save file. Use can use this file whenever you want.

**Lab 18.2: Patching EXEs with Ollydbg**

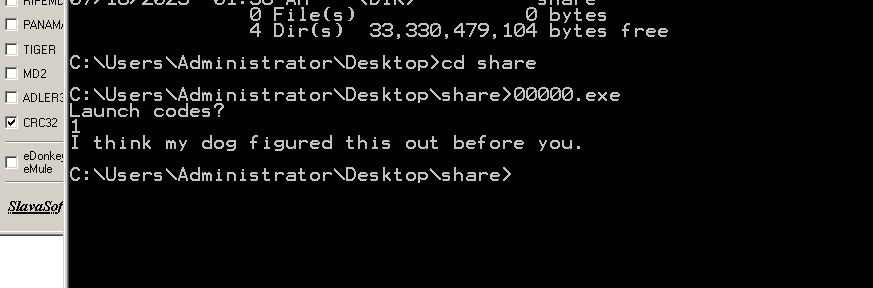
**18.2.1: Patching an EXE**

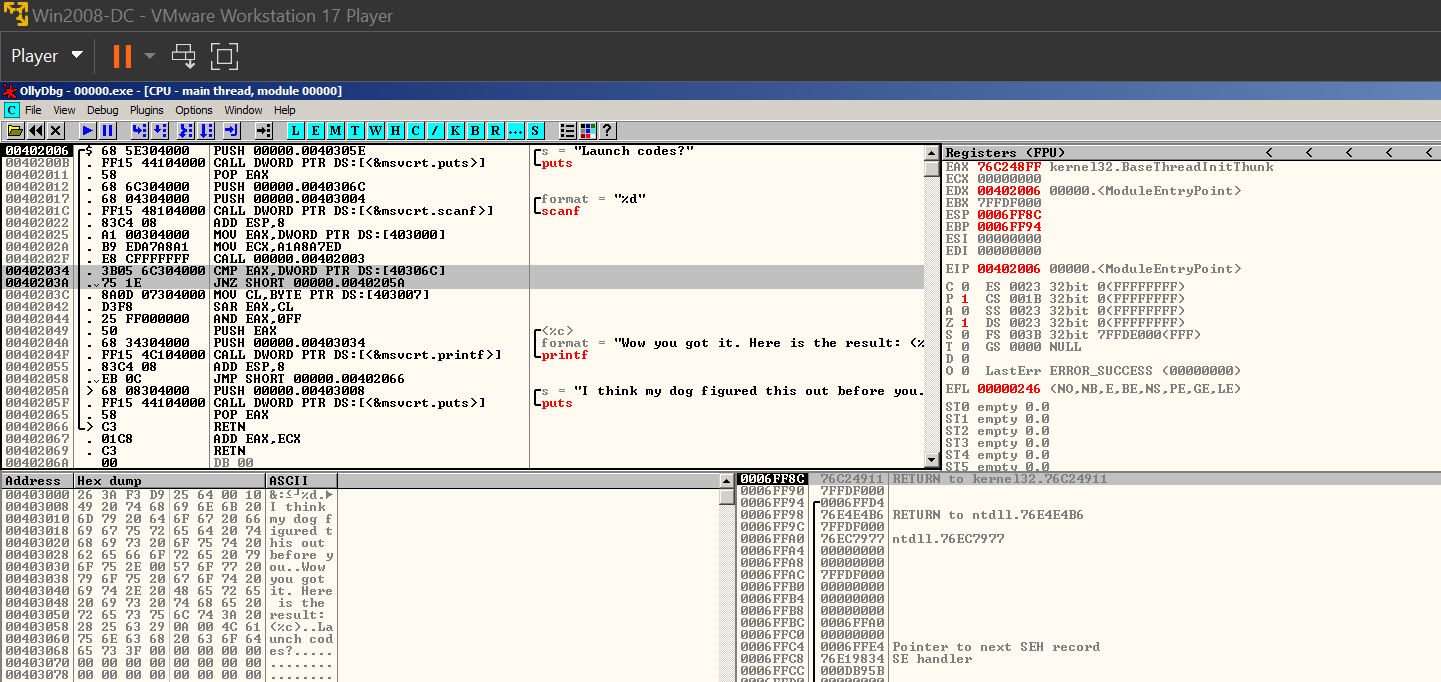
• Checking the Hash: 

• Running the EXE:

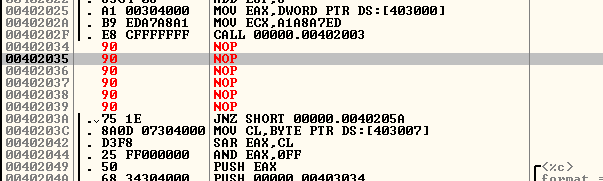
Execute these commands: cd \users\administrator\Desktop

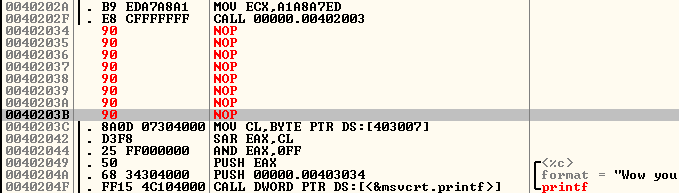
00000.exe

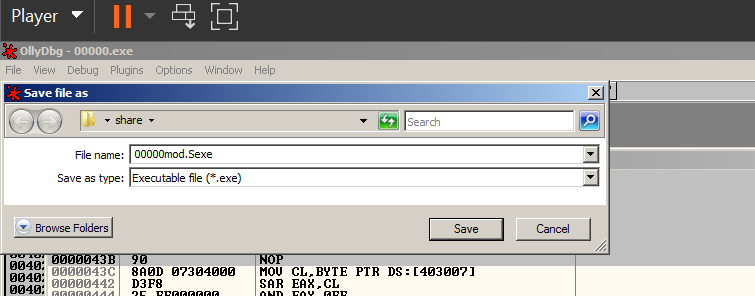
It asks for a "Launch code". Enter 1. Your code is wrong, and it insults you, as shown below

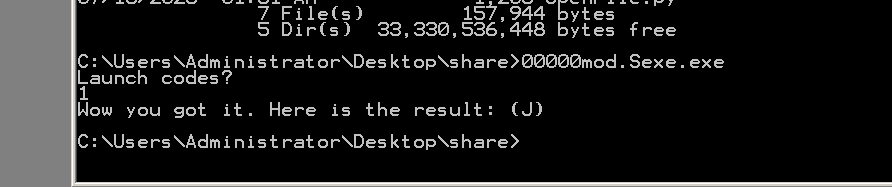
• Examining the EXE with Ollydbg: The choice is performed by two instructions: CMP (Compare) and JNZ (Jump if Not Zero), outlined in green in the image below 

• Modifying the EXE:

Right-click the CMP instruction and click Assemble, as shown below.

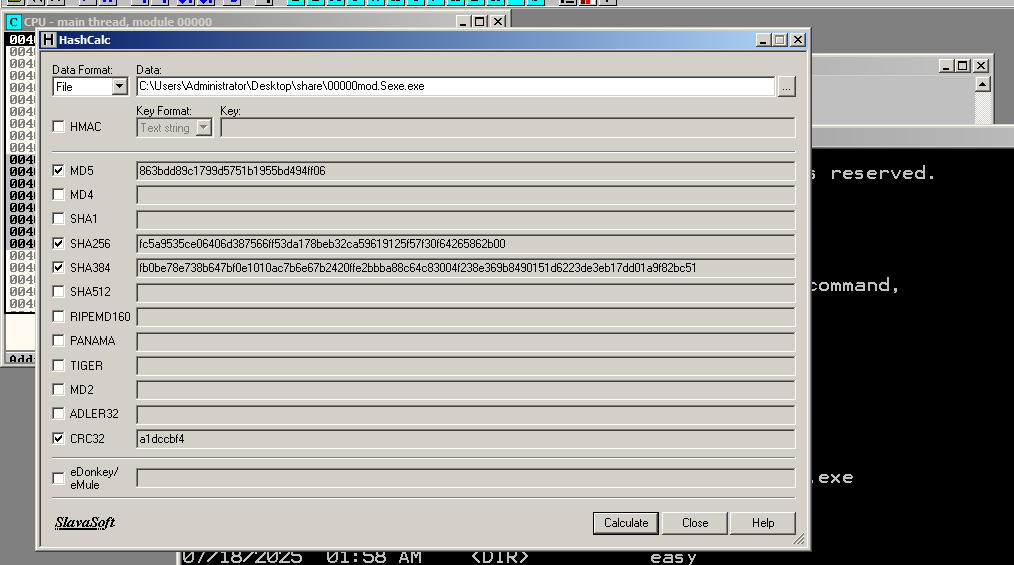
Repeat the process to replace the JNZ instruction with NOPs also, as shown below

* Saving the Modified File: 

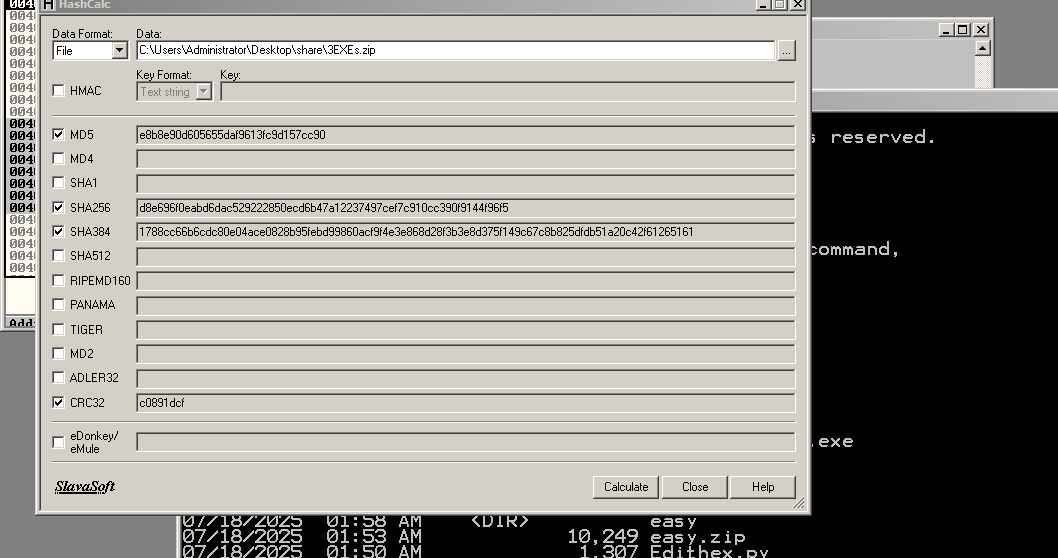
• Running the Modified File: 

• Checking the Hash:

o Calculate the SHA256 hash of the patched file. It should match the value shown below.

o Find the CRC32 hash, which is covered in a green box in the image below. Enter it into the form below

**18.2.2: Patching three EXEs:**

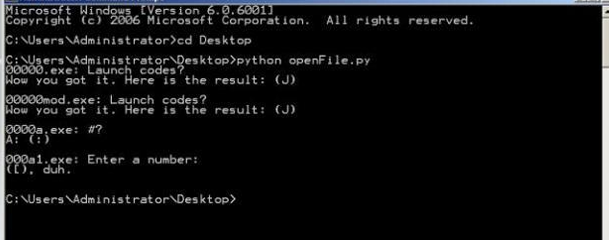
• Checking the Hash: • Patch the Files:

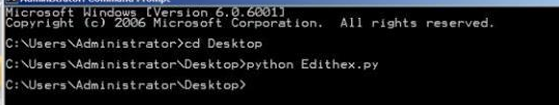
o Patch all 3 files so they will accept any input

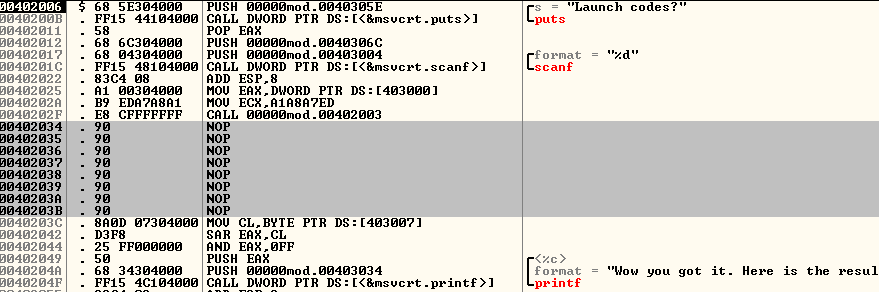
• Gather the Results:

o Run the three patched files. Each one returns a single character as a result. Keep the files in alphabetical order, by filename, like this: File 00000.exe Result C File 0000a.exe Result A

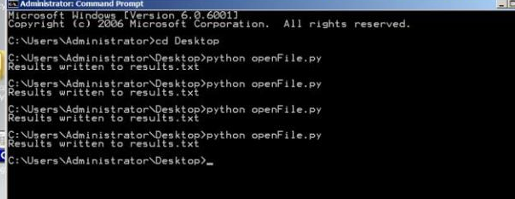
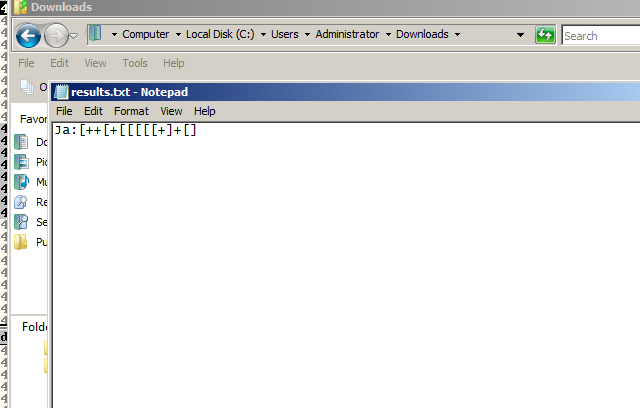
o If those were the results, the answer would be CAT o The actual results are different, of course.



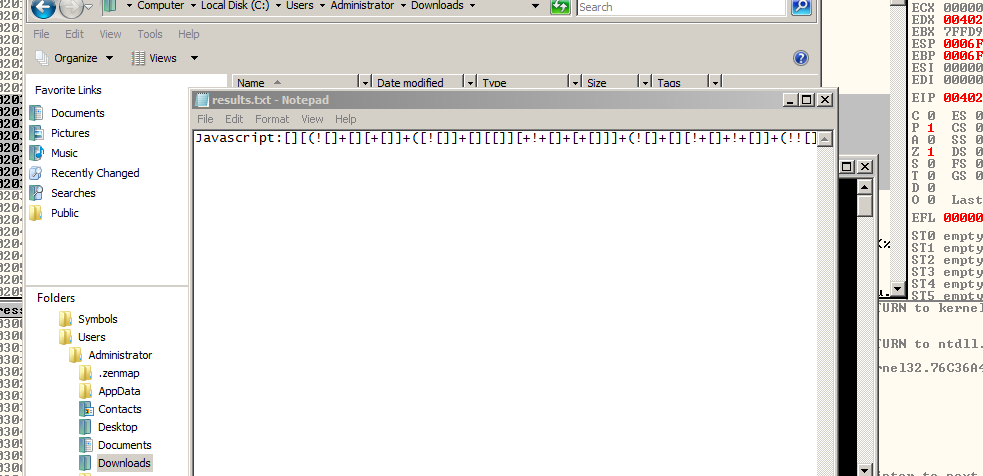
o Now we run python script in cmd: 

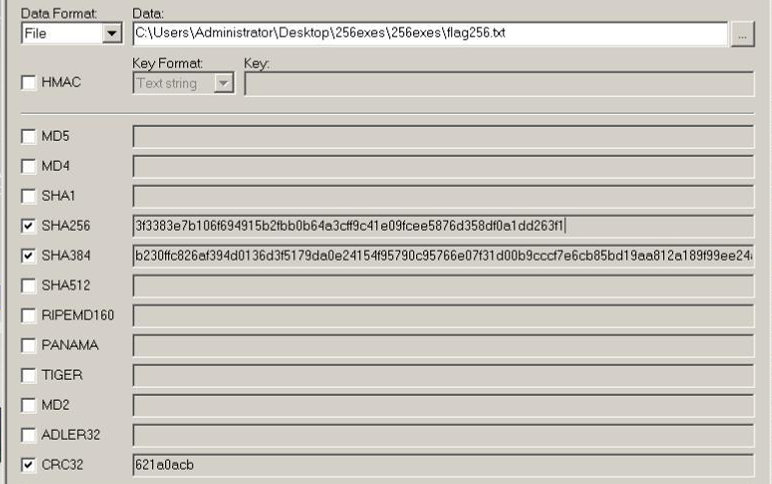
o Check file with Ollydbg

o Now we use another script to run mutilple file and store it in results.txt

**18.2.3: Patching 19 EXEs:** ****

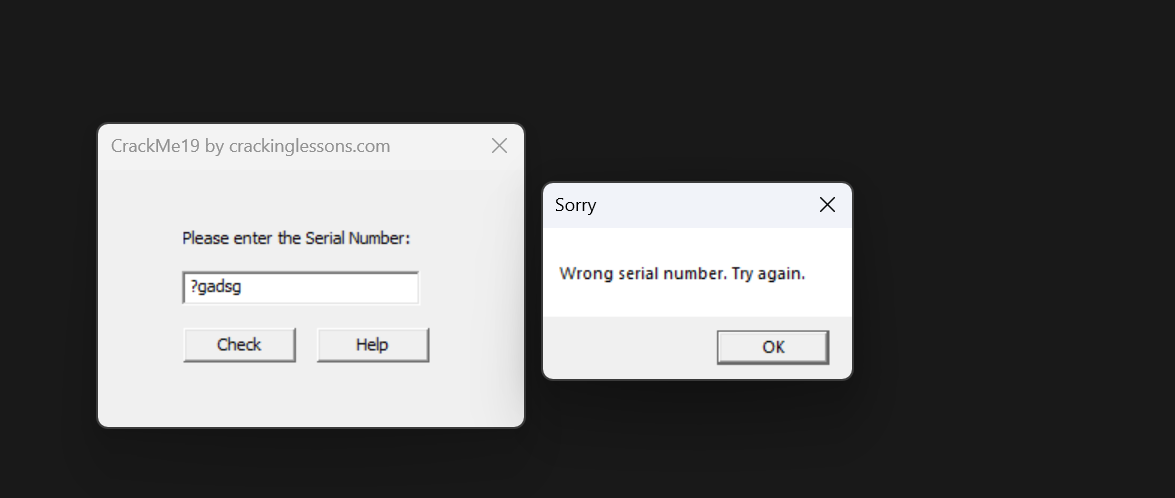
**18.2.4: Patching 256 EXEs:**

o Now we got the string as below

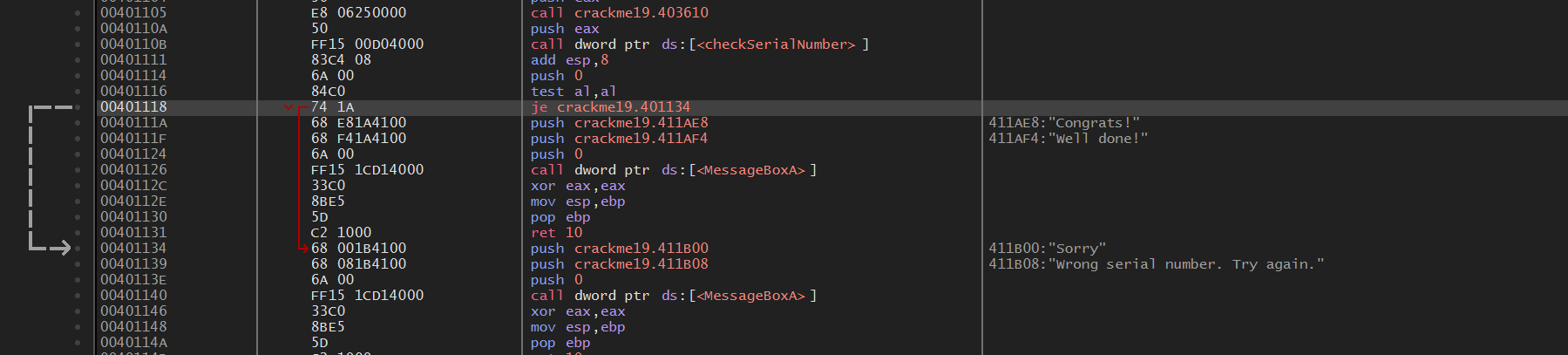
o Calculate the SHA256 hash of that file. 

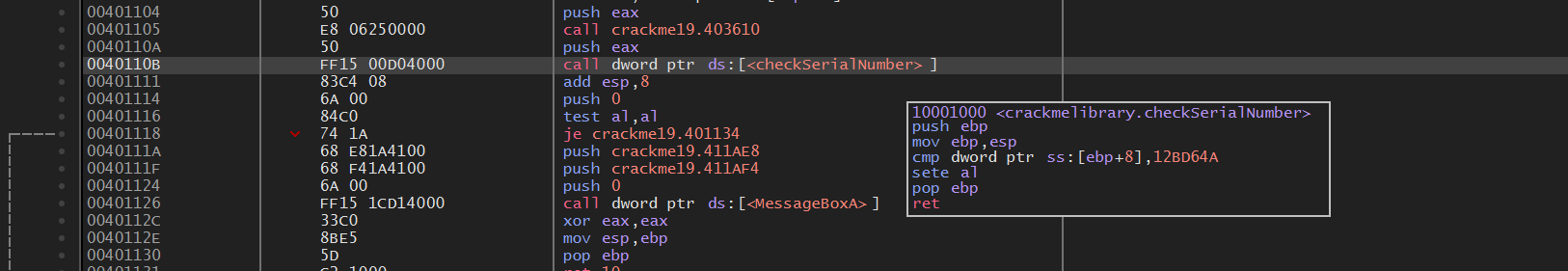
**CRACKME 19**

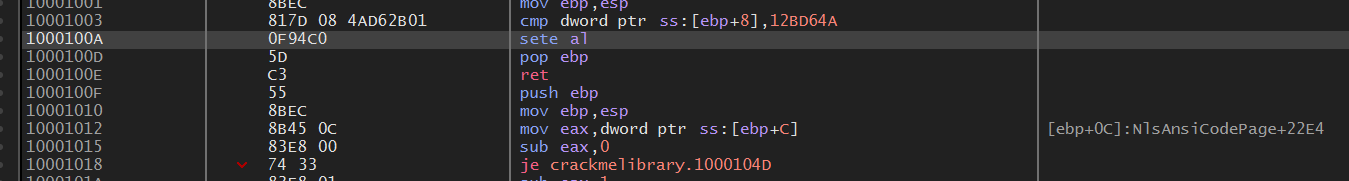
This crackme comes in 2 files. The crackme19.exe is the main file and there is also a DLL called CrackmeLibrary.dll. The objective of this crackme is to practise patching the DLL instead of the crackme19.exe file.

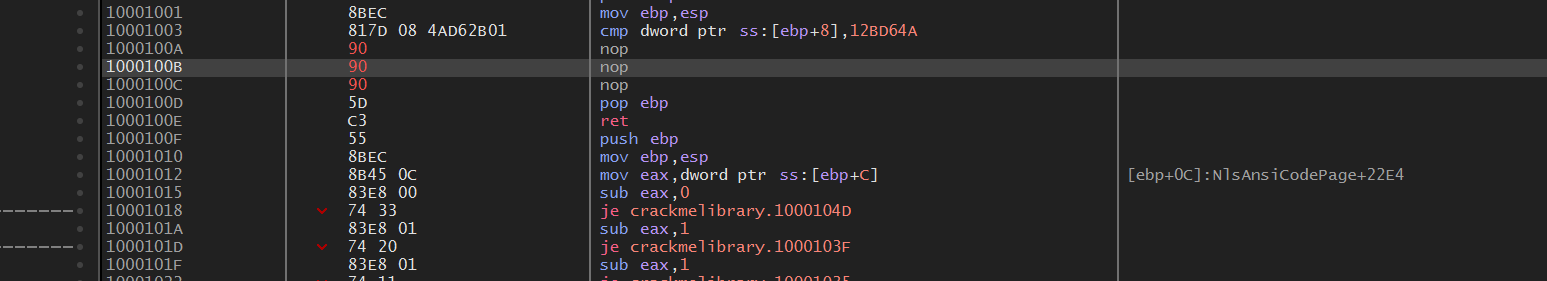


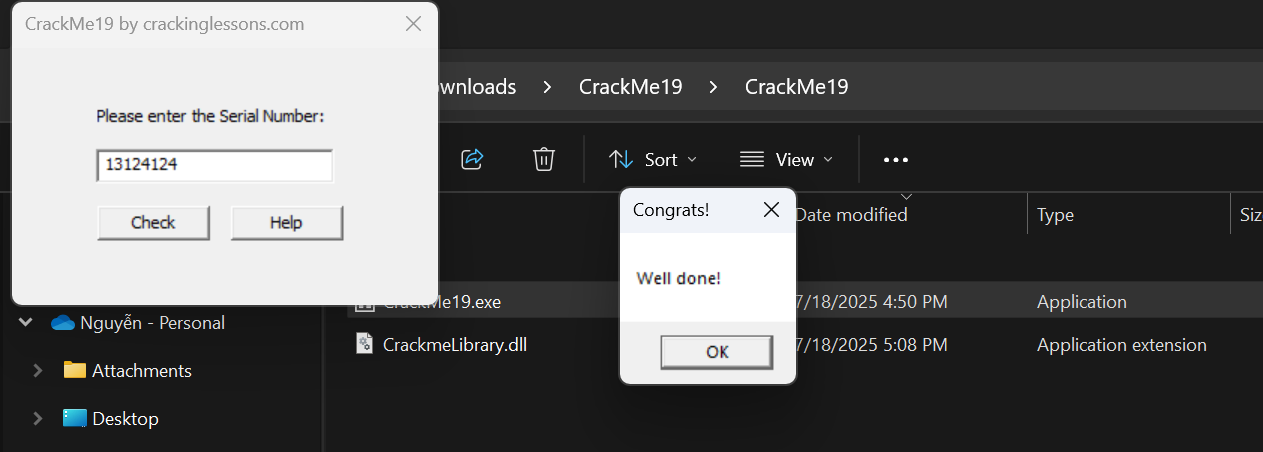
Becasue this challenge provides .dll file, first we need to find string references their code of success part and fail part, and notice this JE command. Above these command is the TEST command for AL variable. If AL equals 0, the JE will be executed. So we need to change the value of AL instead of 0 value.



Notice the function checkSerialNumber, this function is block assigning AL with 0 value.

Inspect into this function, we found the code SETE AL, this assigning 0 into AL, so we need to fill NOP in this place.

Like below:

After that we patch the file, but notice that, function checkSerialNumber is executed in .dll file, so we need to patch this file into new .dll replacing the old one.

Done!!!