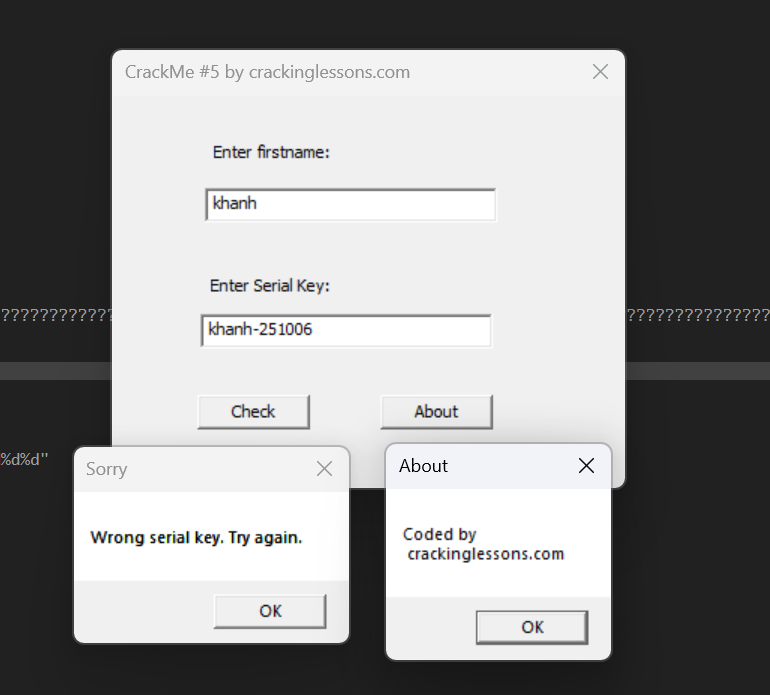
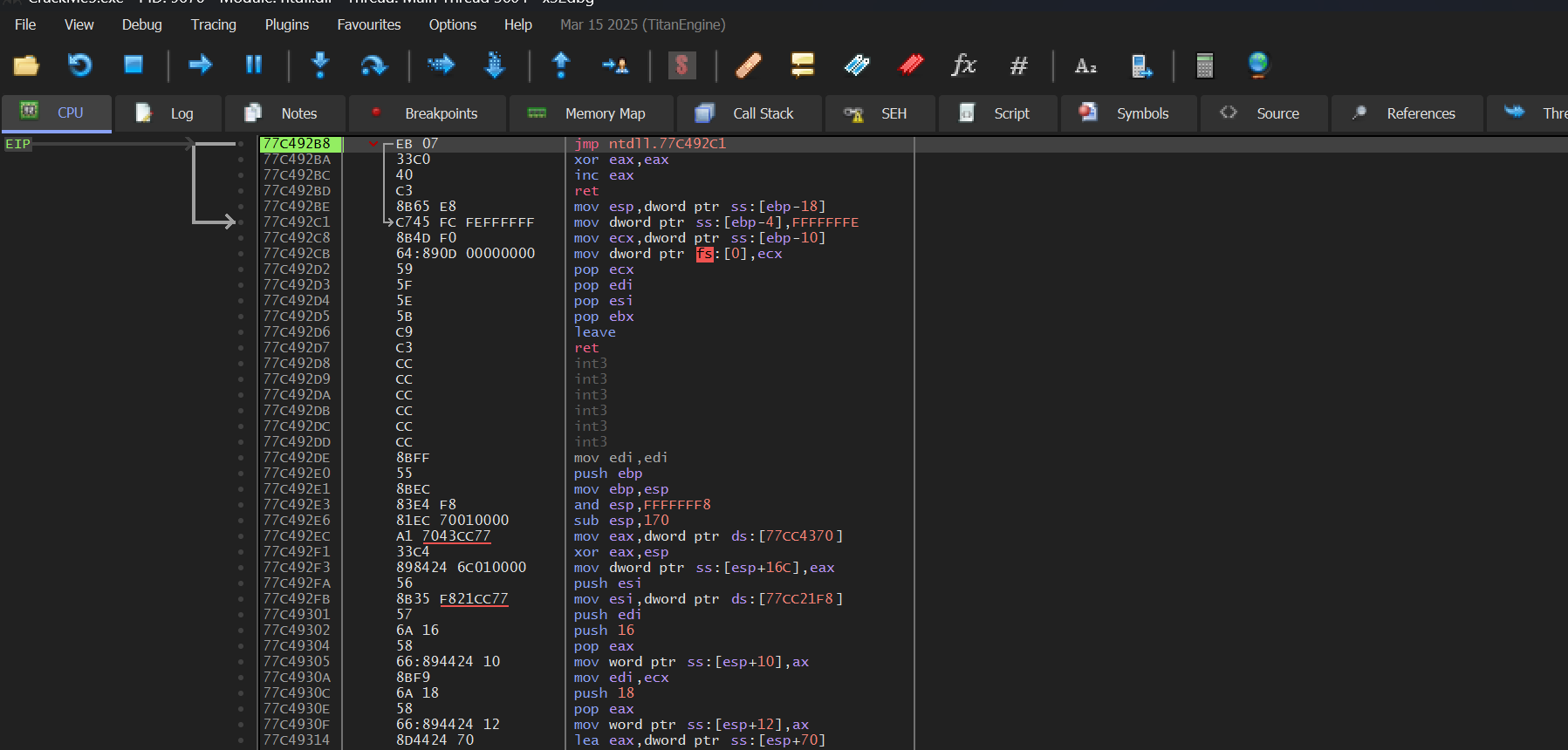
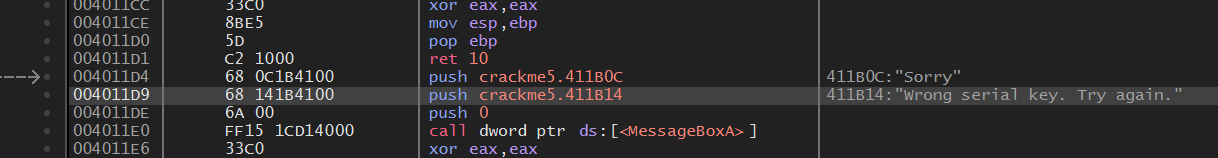
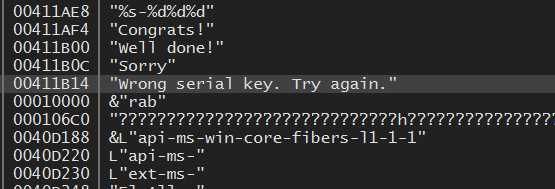
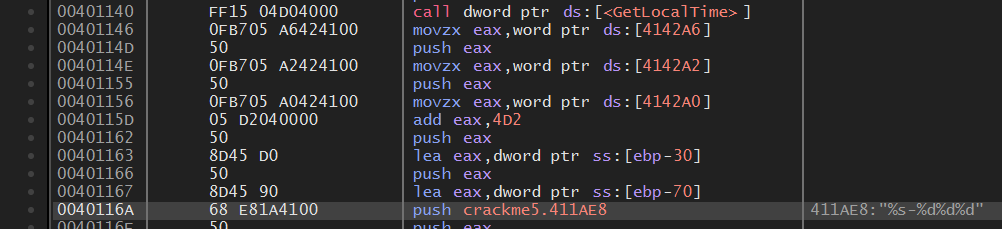
CRACK ME 5



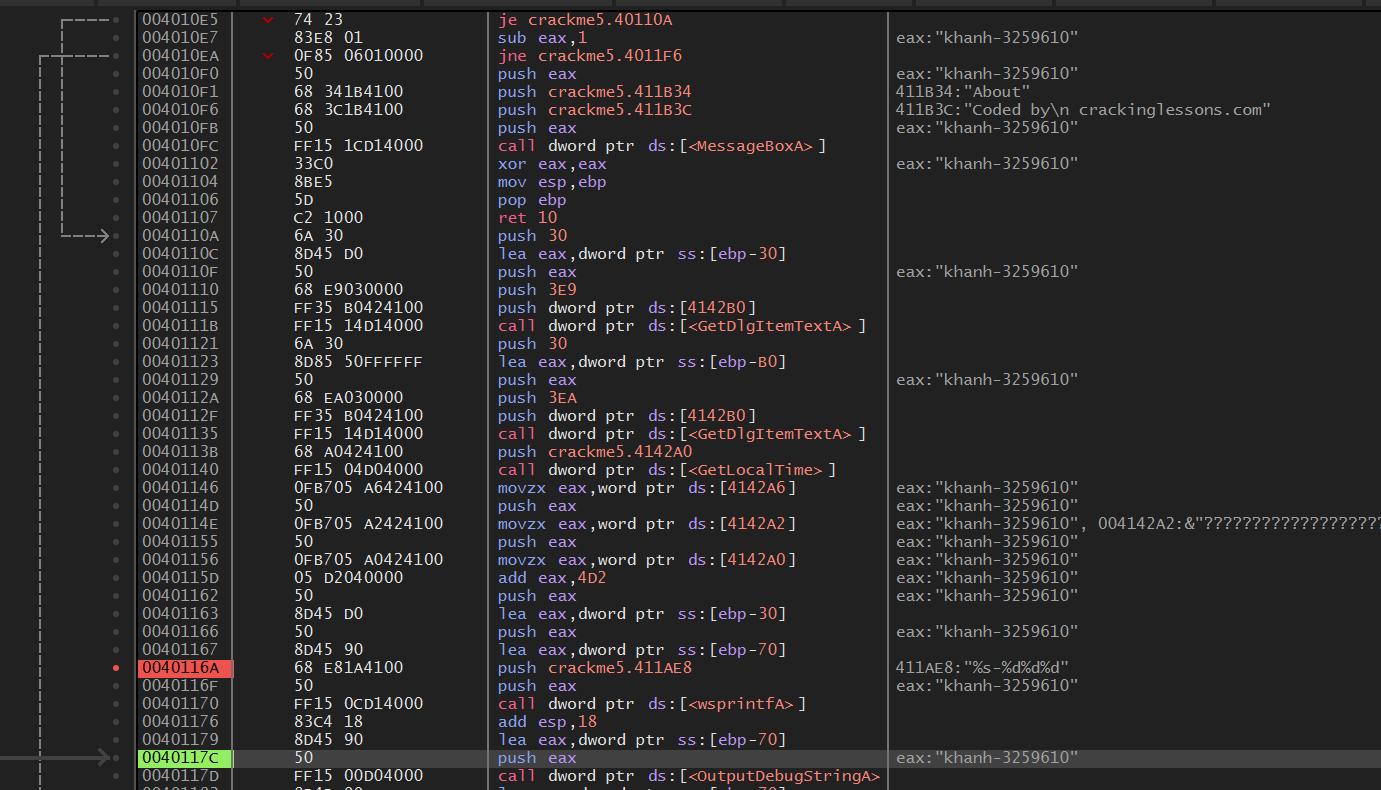
Inspect this file .exe in x32dbg tool.

Find string reference to string “Wrong serial key. Try again.”. Click to jump to the line of code of these string.



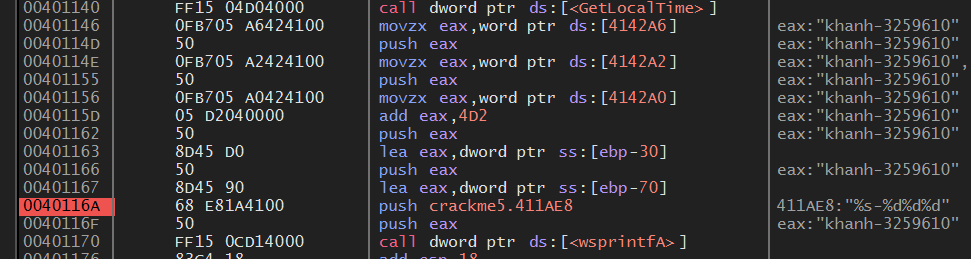
Trace back to the above code to see what operates, leading to this “Sorry” code. Notice to this code part, from function GetLocalTime, analysing variabe EAX. May be, EAX represents for serial key. But now, ignoring it for a bit, we add breakpoint to code line of function GetLocalTime and step into to deeply analyse.

After many times for tracing and putting breakpoint, I get some detail to see.

At the address 0040116A, we have the comment: “411AE8: “%s-%d%d%d””, below this is hint “khanh-325910”, this is likely to be serial key. 

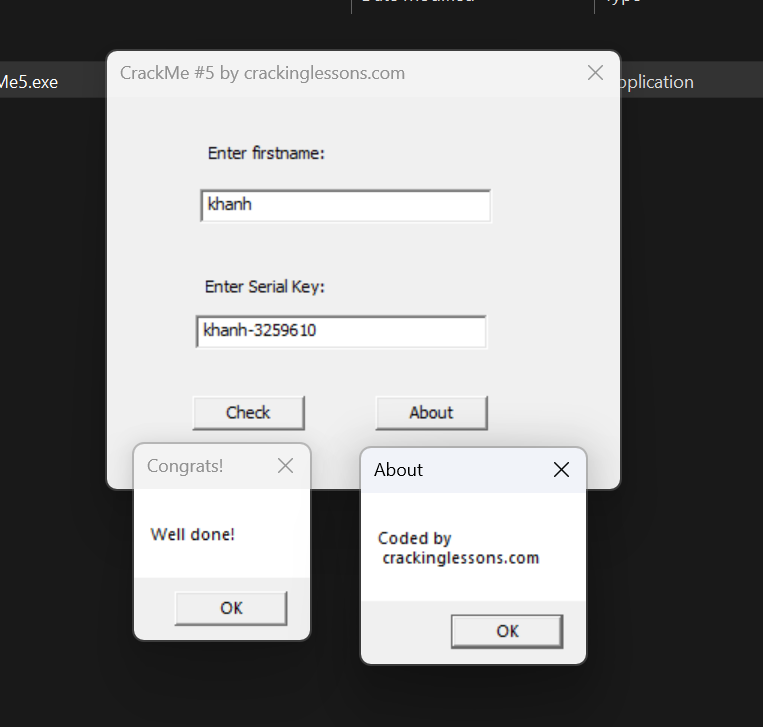
But why we have this form of serial key? The regex “%s-%d%d%d” equal to khanh-325910.

%s is for khanh (the firstname) combine with -%d%d%d to -3259610.

Let trace back again to this part of code below: 

System call GetLocalTime this means get the time schedule of system, at this time is 10th of june 2025. If mapping to regex, we have -2025610, but serial key is -3259610.

Notice to the address 0040115D, after assigning and pushing numbers of month and day to EAX. System assigns number of year to EAX, and at this line of code, adding EAX with 4D2 in hexa (1234 in decimal). 2025 adds with 1234 euqal to 3259. Hence, we have Serial Key: khanh-3259610.



Typing firstname and try the cracked serial key to check.

DONE!!!