THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

Trần Đại Chí - 18127070

Trần Huy Vũ - 18127257

ĐỀ TÀI

Crack

Chuyên ngành: Công nghệ thông tin

Môn: Kiến trúc máy tính và hợp ngữ

Thành phố Hồ Chí Minh - 2019

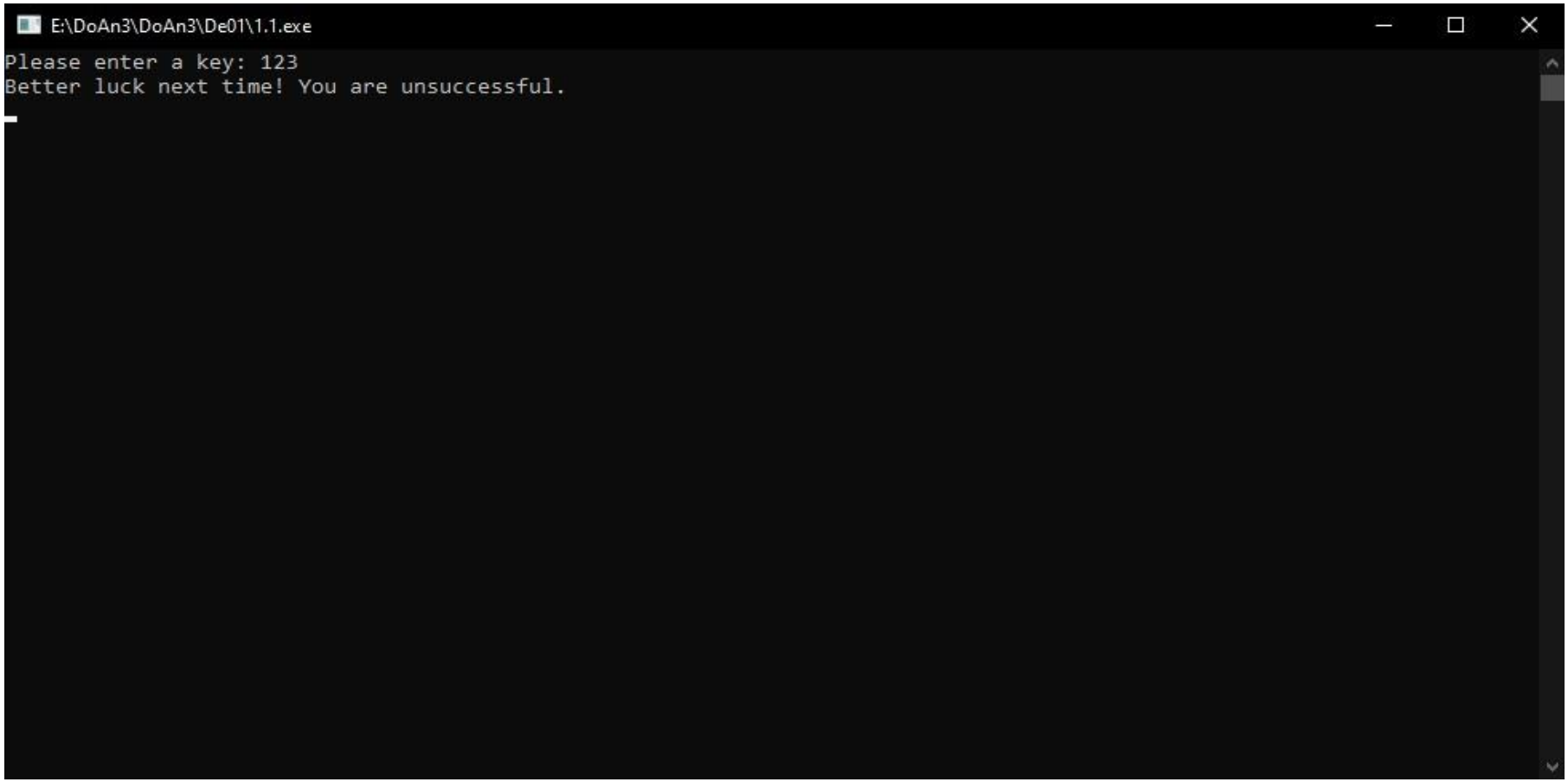
THÔNG TIN THÀNH VIÊN NHÓM

STT	Họ tên	MSSV
1	Trần Đại Chí	18127070
2	Trần Huy Vũ	18127257

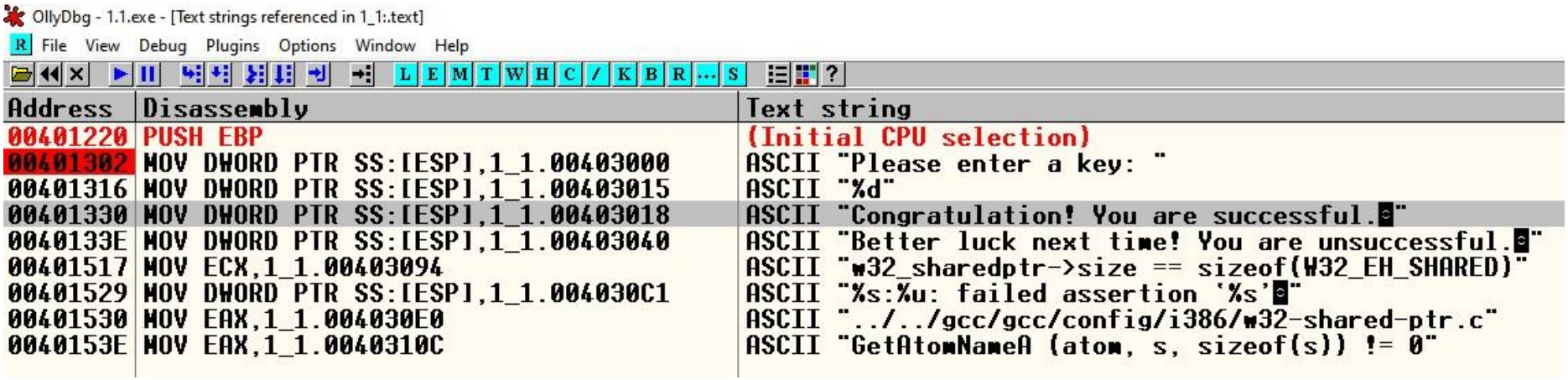
I. BẢNG PHÂN CÔNG CÔNG VIỆC

Trần Đại Chí	1.2, 1.5, viết keygen
Trần Huy Vũ	1.1, 1.3, 1.4

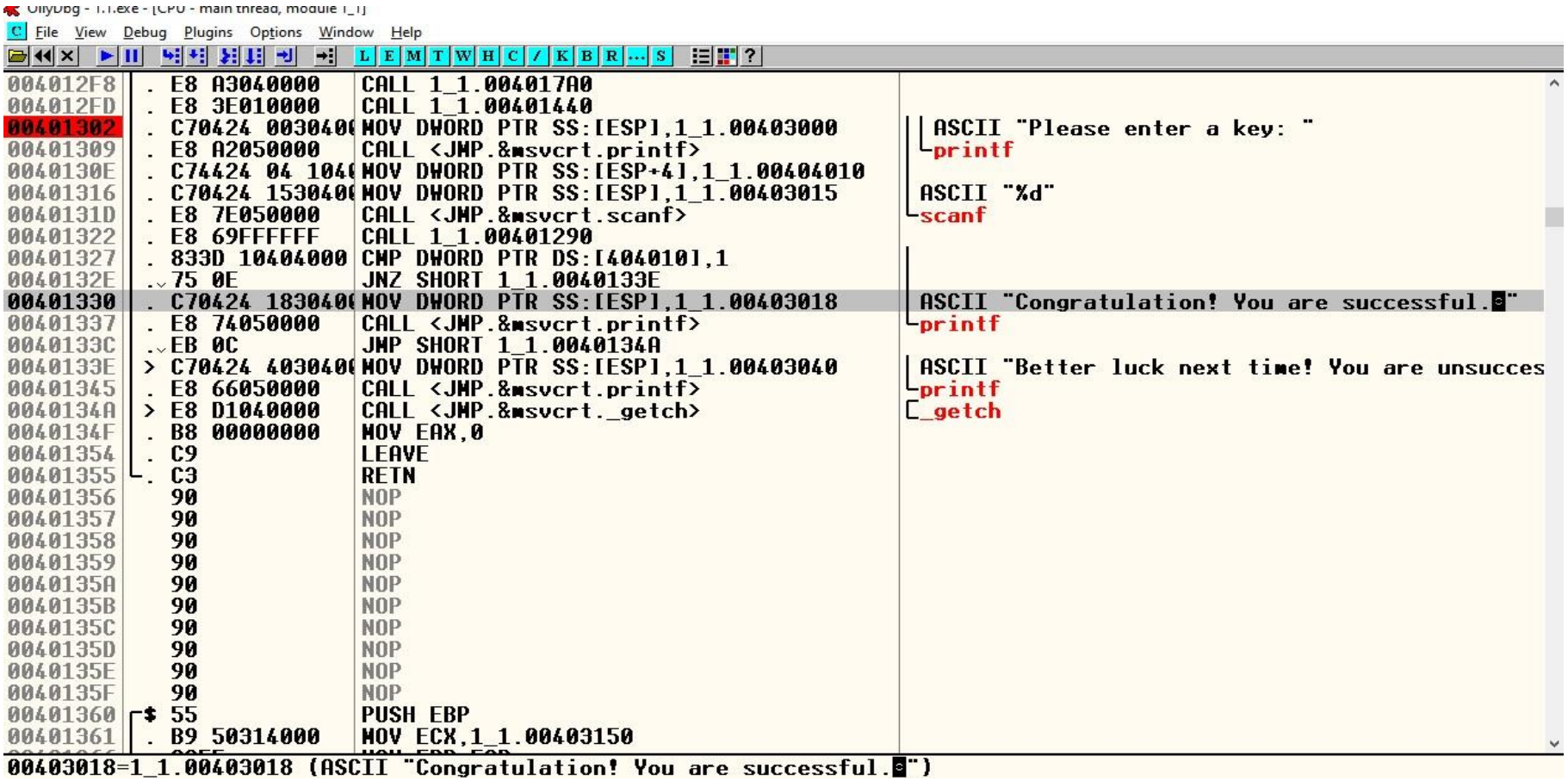
1.1



Đầu tiên ta chạy chương trình và nhập vào 1 số bất kì chương trình sẽ hiện thông báo “Better luck next time! You are unsuccessful”



Sau đó ta mở OllyDbg, mở 1.1.exe, chuột phải chọn search for -> All preferenced text strings, ta thấy được hai chuỗi hiện ra thông báo khi ta nhập key đúng và khi nhập sai. Sau đó nhấn vào đi đến đoạn mã chưa chuỗi đó

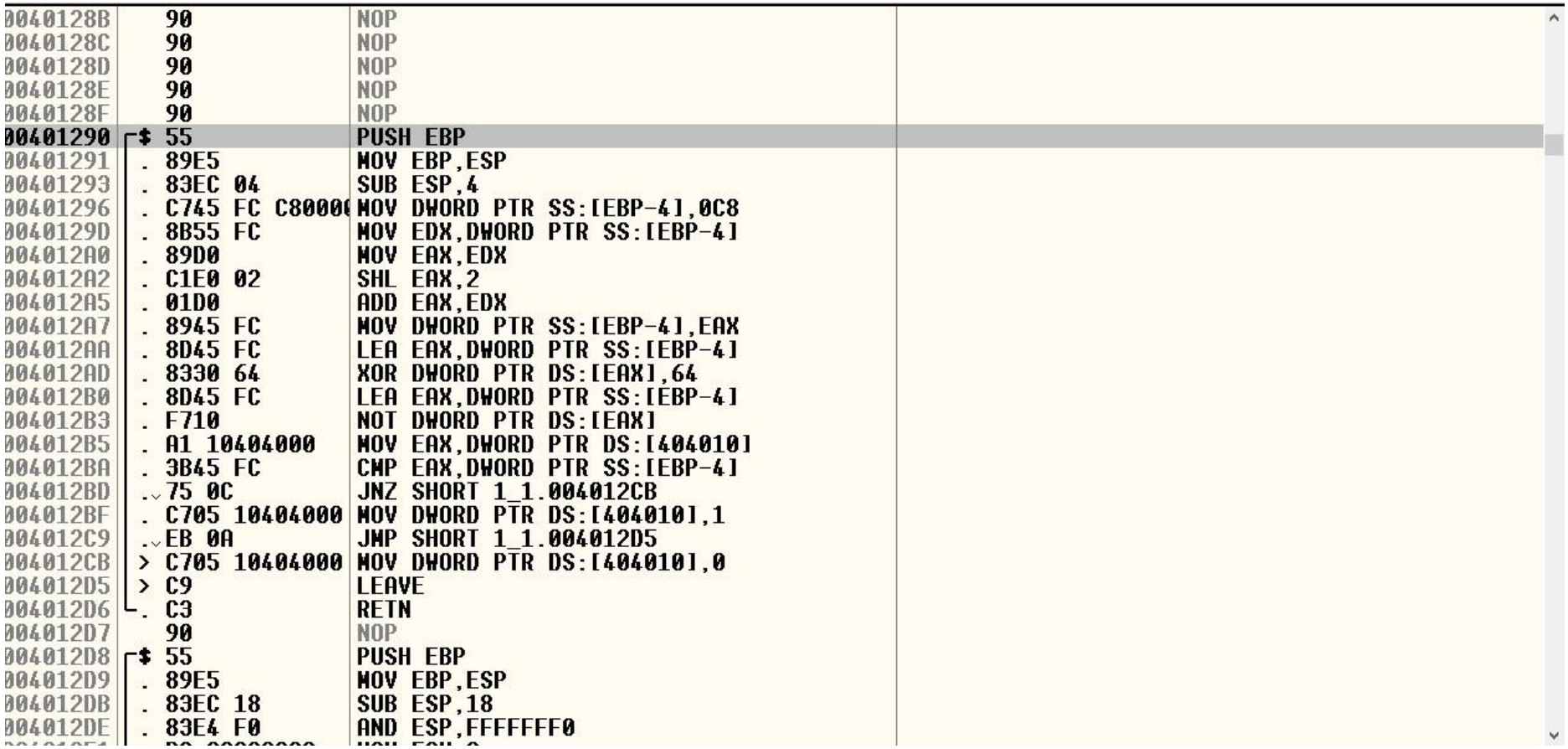


00401316 ASCII “%d”

0040131D scanf

Hai dòng trên gọi hàm scanf yêu cầu người dùng nhập key vào

00401322 CALL 1_1.00401290: đi tới dòng lệnh có địa chỉ 00401290



Ta đi tới dòng lệnh tương ứng địa chỉ đó 00401296 MOV DWORD PTR SS:[EBP-4].0C8: lưu giá trị 0C8 vào địa chỉ EBP-4

0040129D MOV EDX.DWORD PTR SS:[EBP-4]

004012A0 MOV EAX.EDX

Hai dòng trên có nghĩa lưu giá trị của EBP-4 vào thanh ghi EDX rồi EAX = EDX = 0C8

004012A2 SHL EAX,2: dịch trái EAX 2 lần

004012A5 ADD EAX,EDX: EAX = EAX + EDX = 3E8

004012A7 MOV DWORD PTR[EBP-4], EAX: lưu EAX vào EBP-4

004012A2 LEA EAX, DWORD PTR[EBP-4]: gán EAX vào EBP-4 (EBP = 0023FF08) -> EAX = 0023FF04

004012AD XOR DWORD PTR[EAX], 64: phép XOR EAX với 64 -> 38C

004012B3 NOT DWORD PTR[EAX]: phép NOT với EAX -> FFFFFC73, đổi ra decimal ta được kết quả là -909

004012B5 MOV EAX, DWORD PTR[404010]: gán EAX bằng giá trị vùng nhớ có địa chỉ 404010 -> EAX = 4D2

004012BA CMP EAX, DWORD PTR[EBP-4]

004012BD JNZ SHORT 1_1.004012CB

004012BF MOV DWORD PTR[404010], 1

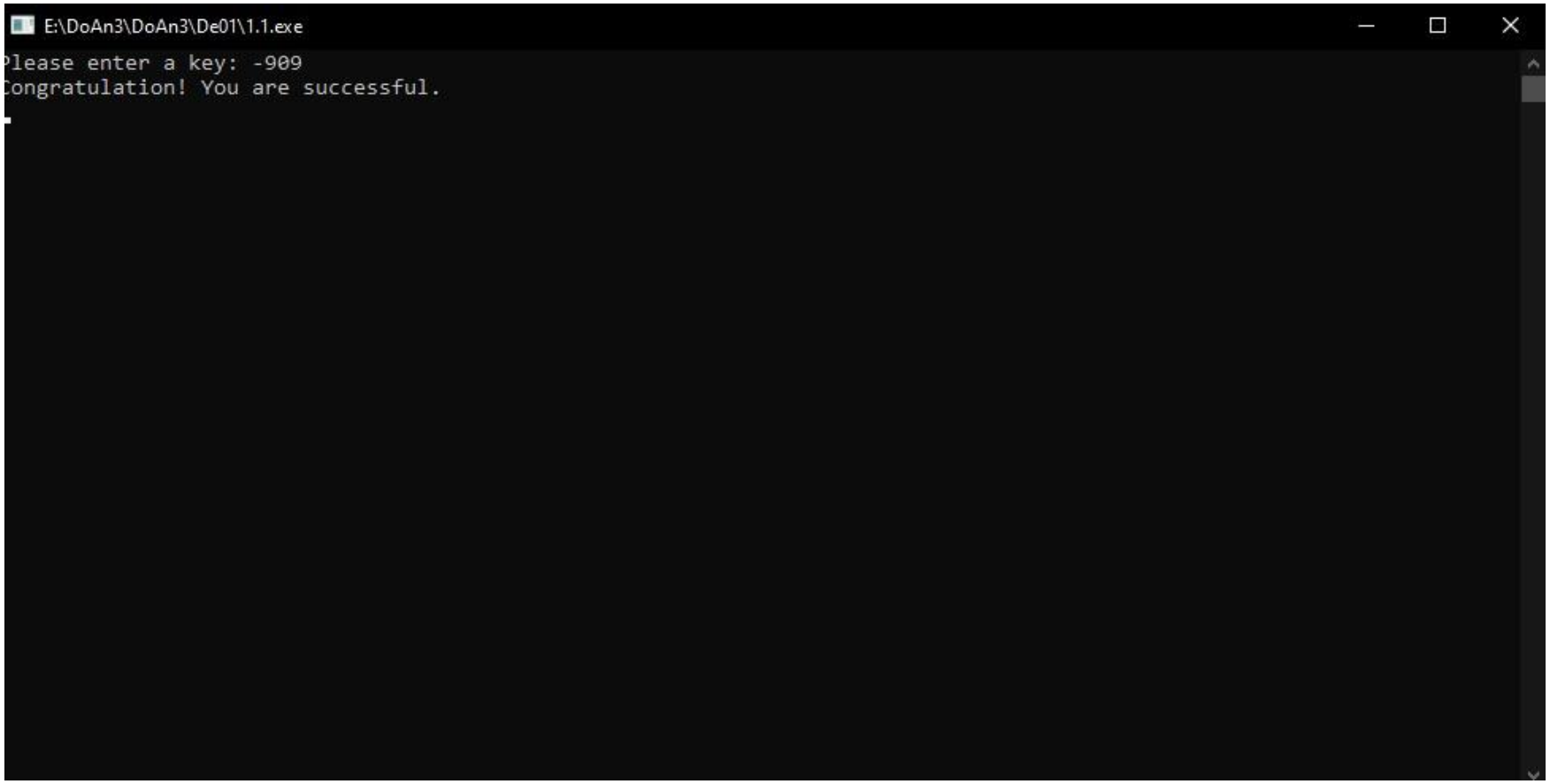
004512C9 JMP SHORT 1_1.004012D5

004012CB MOV DWORD PTR[404010], 0

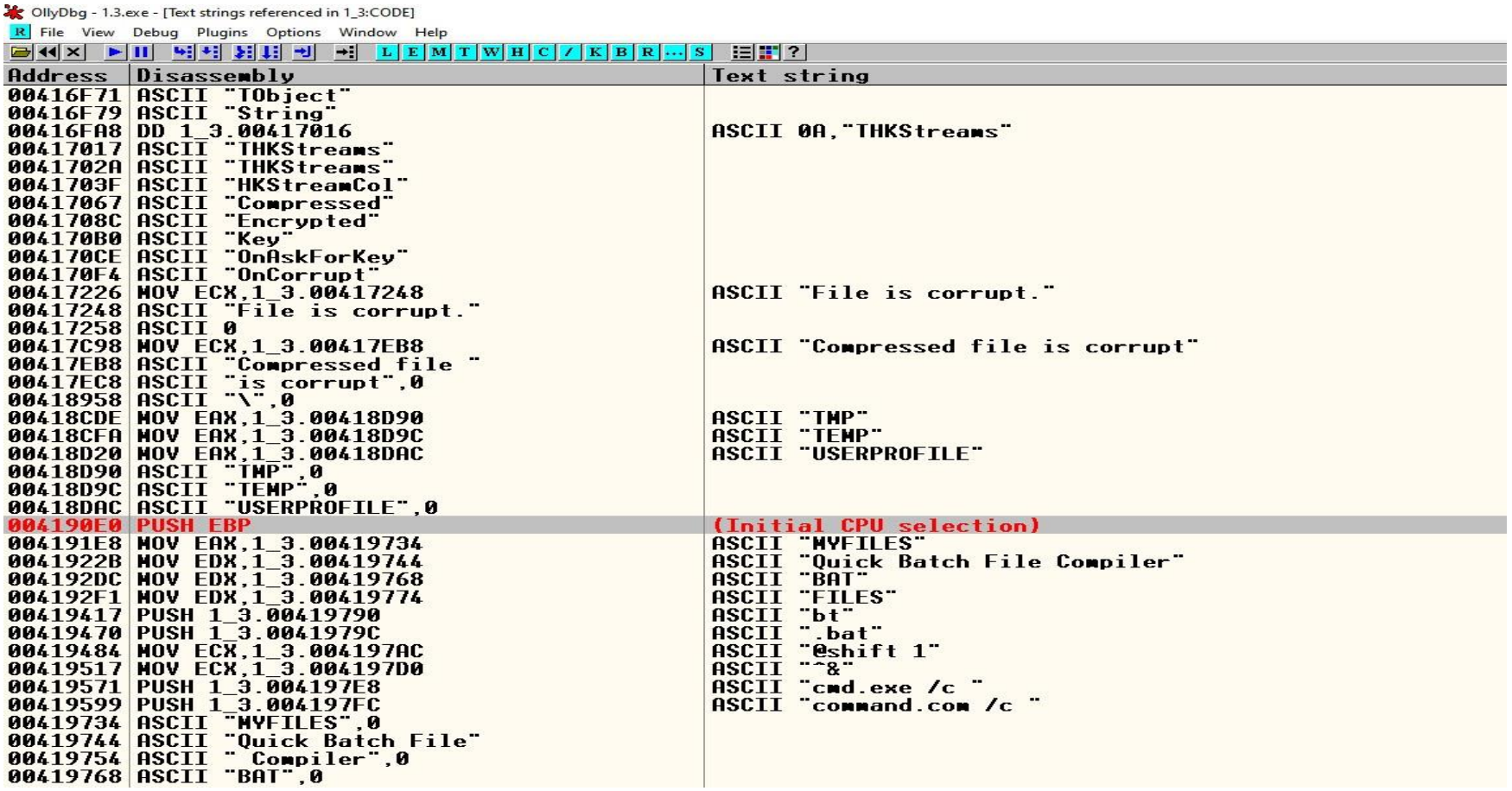
004012D5 LEAVE

004012D6

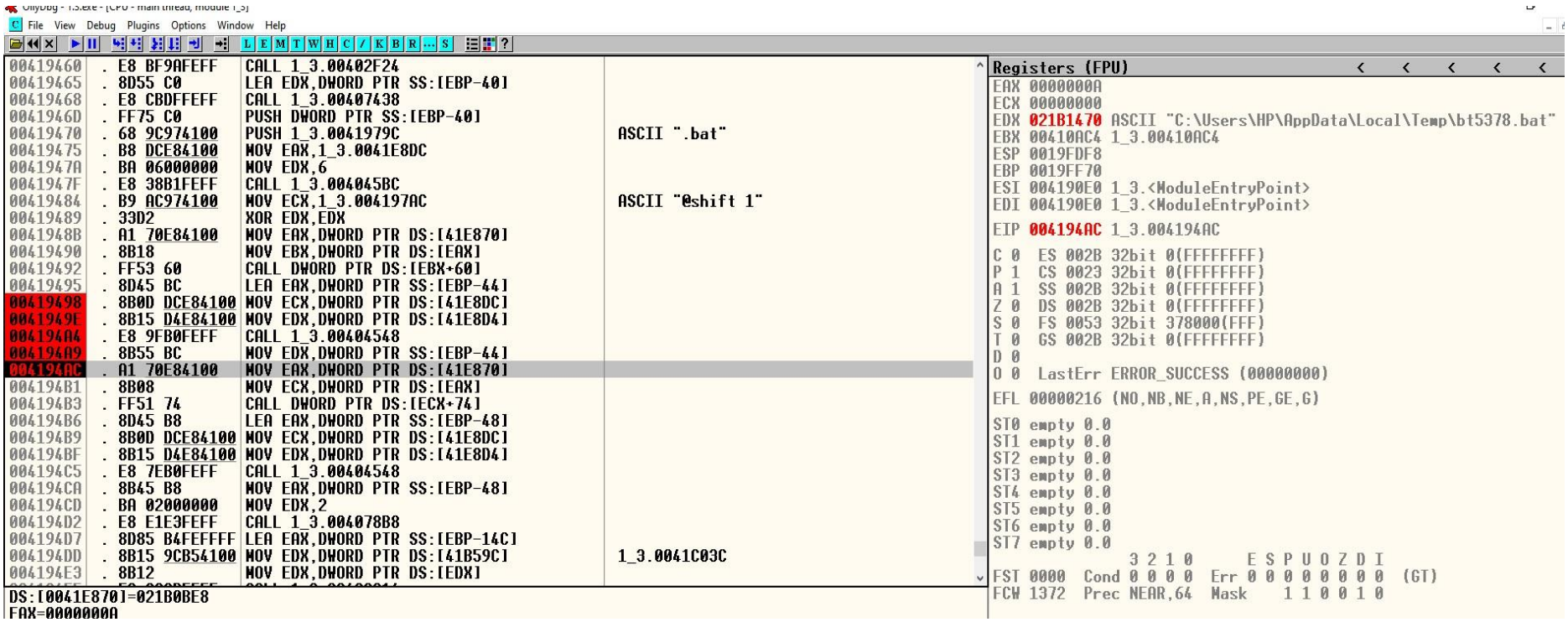
So sánh EAX với EBP-4(= -909). Nếu bằng thì gấn giá trị ở vùng nhớ 404010 bằng 1, ngược lại gán bằng 0. Sau đó quay về dòng lệnh 00401327 (nếu bằng 1 thì in ra thông báo successful, ngược lại bằng 0 thì là unsuccessful). Như vậy key cần tìm ở đây là -909


E:\DoAn3\DoAn3\De01\1.1.exe
Please enter a key: -909
Congratulation! You are successful.

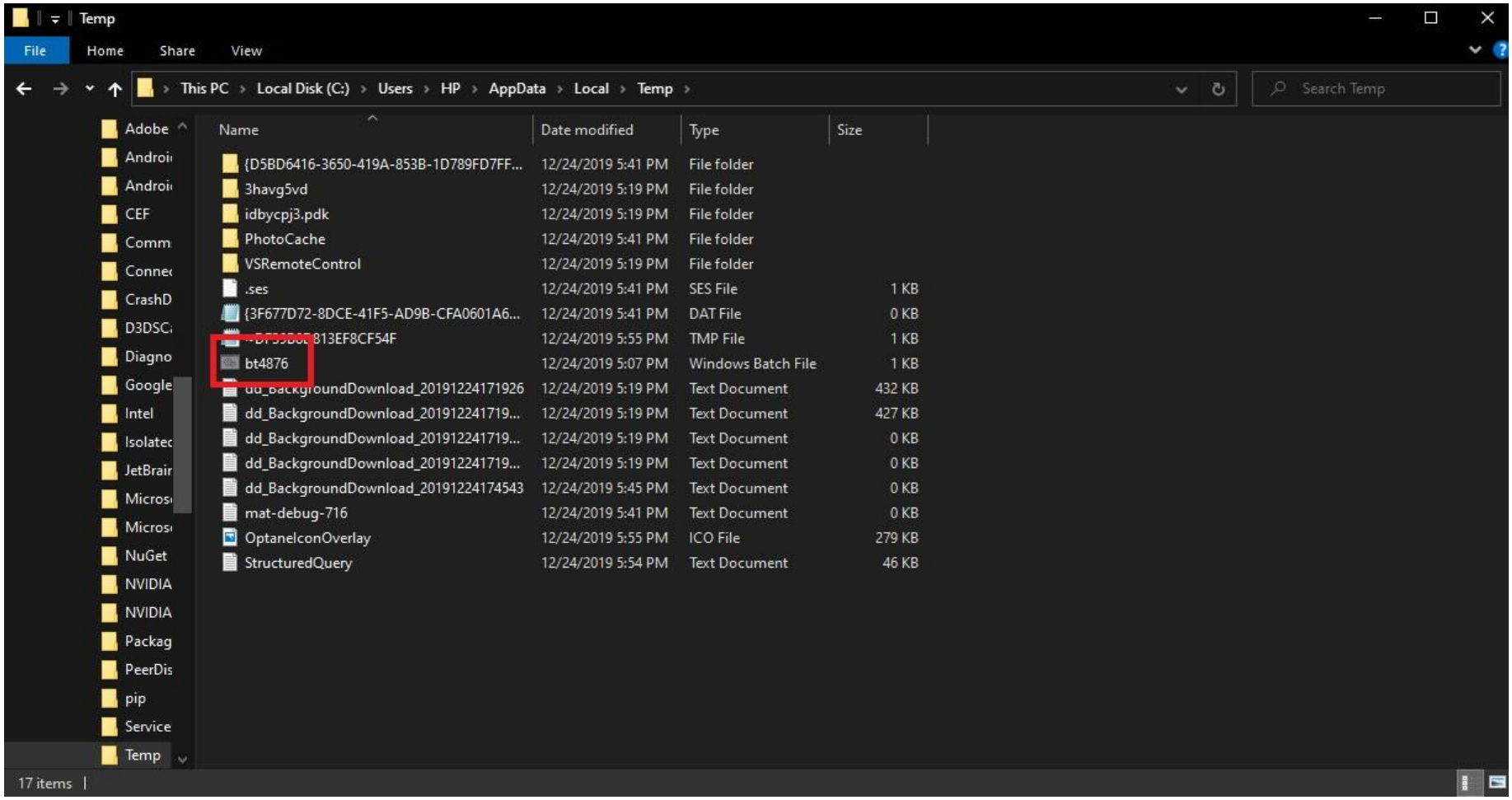
1.3



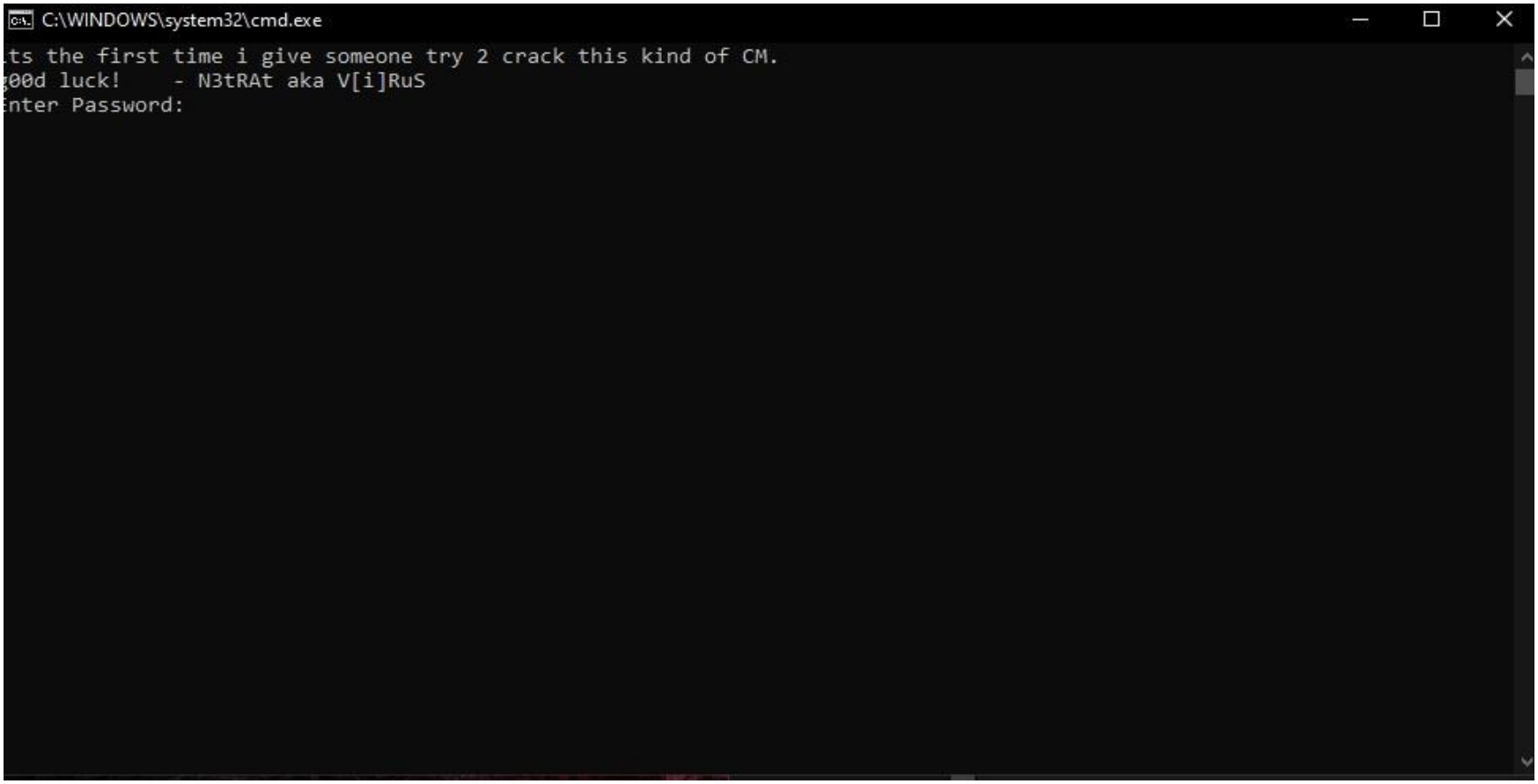
Khi mở chương trình lên, tìm trong chương trình ta không thấy đoạn string khi mở chương trình lên để nhập password -> vùng nhớ không thể truy xuất khi chương trình chưa chạy



Ta tiến hành debug và thấy được một đường dẫn xuất hiện ở 5 dòng sau “C:\HP\AppData\Local\Temp\bt5378.bat”

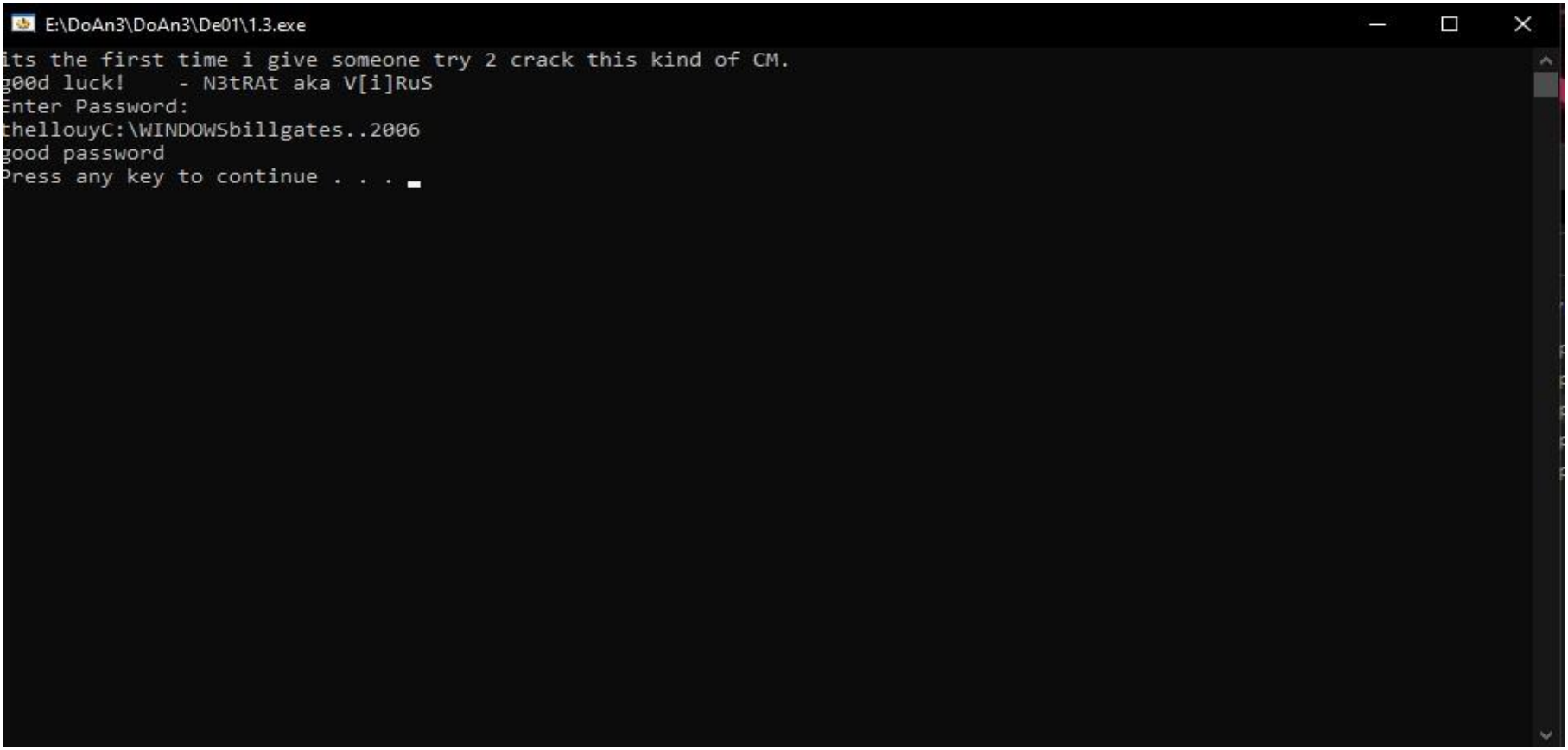


Ta tiến hành mở đường dẫn và có một file bt4876.bat như trong chương trình



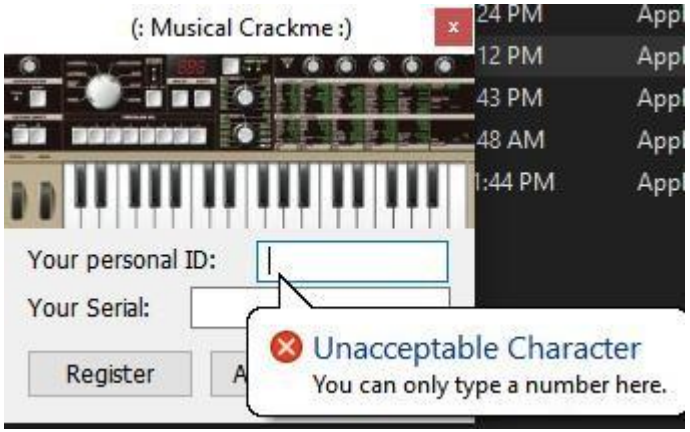
Ta tiến hành mở file lên và thấy xuất hiện những dòng như giống như chương trình chính -> ta có thể đoán được file .bat này hoạt động tương tự như chương trình chính nên ta tiến hành edit file .bat này để có thể thấy được password của nó

Sau khi ta edit file này và có được password của chương trình là thellouyC:\WINDOWSbillgates..2006



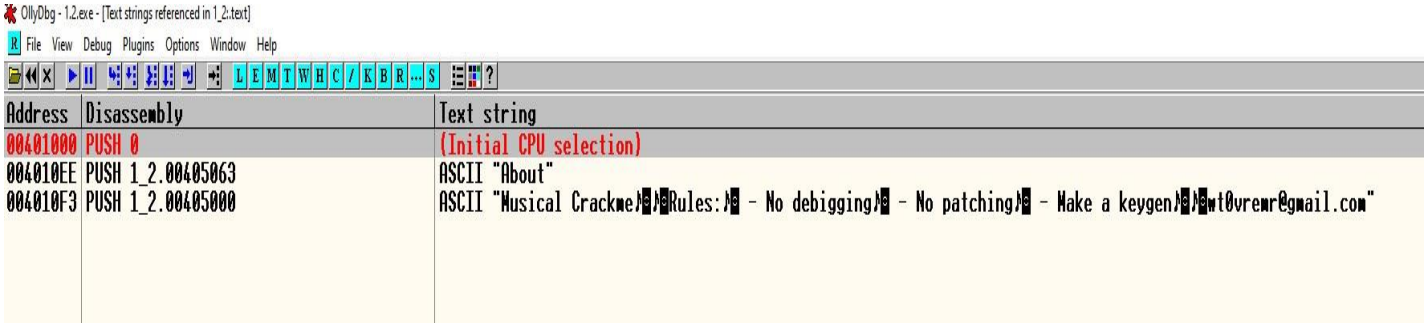
Ta thử lại chương trình với password trên và kết quả là thành công

1.2

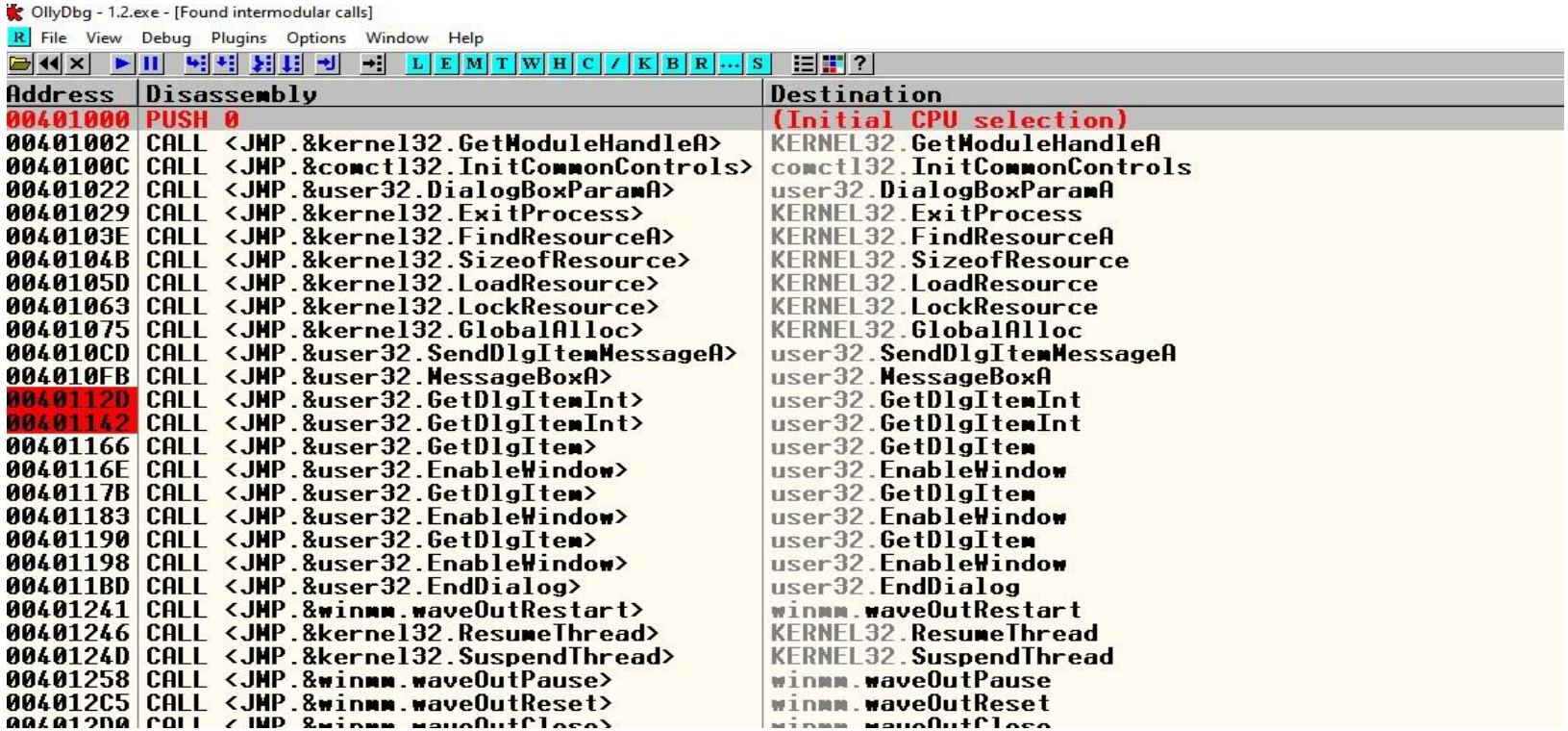


Đầu tiên mở chương trình lên ta nhập thử số vào ID thì thấy báo lỗi -> ID chỉ có thể nhập số

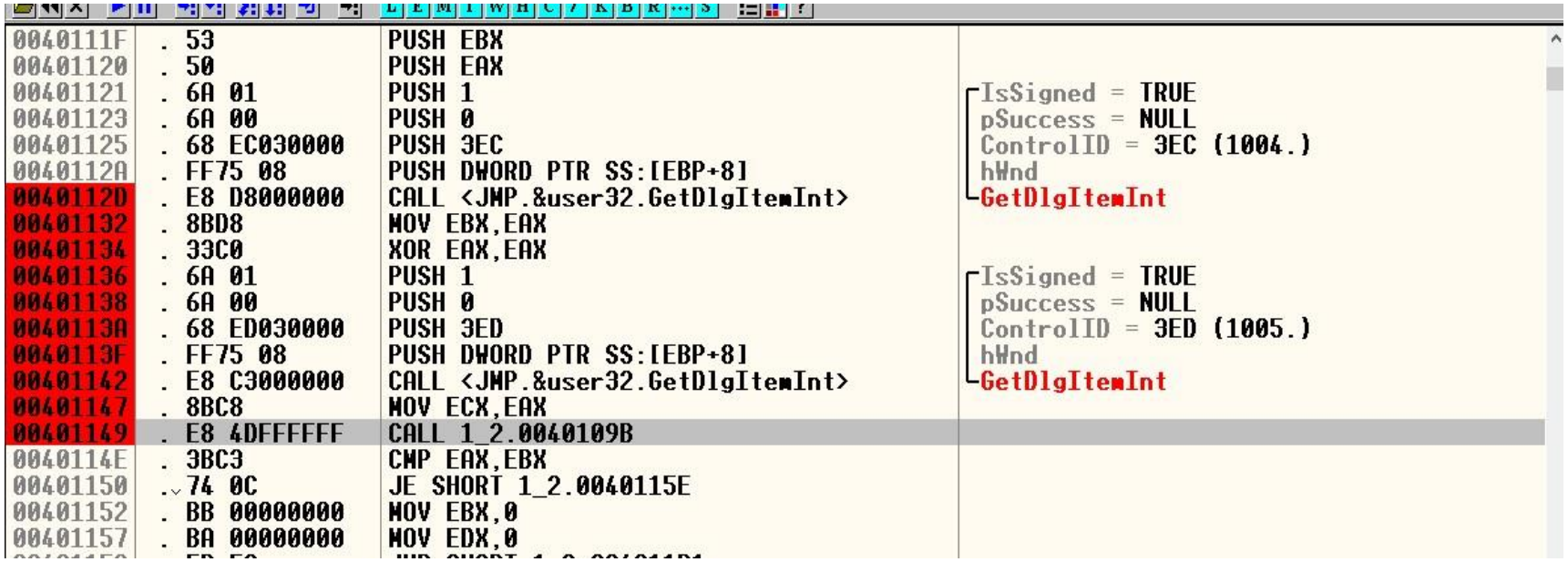
Tiếp đó đến nhập serial ta thấy chỉ có thể nhập tối đa là 16 chữ số -> không nhập gì cũng được



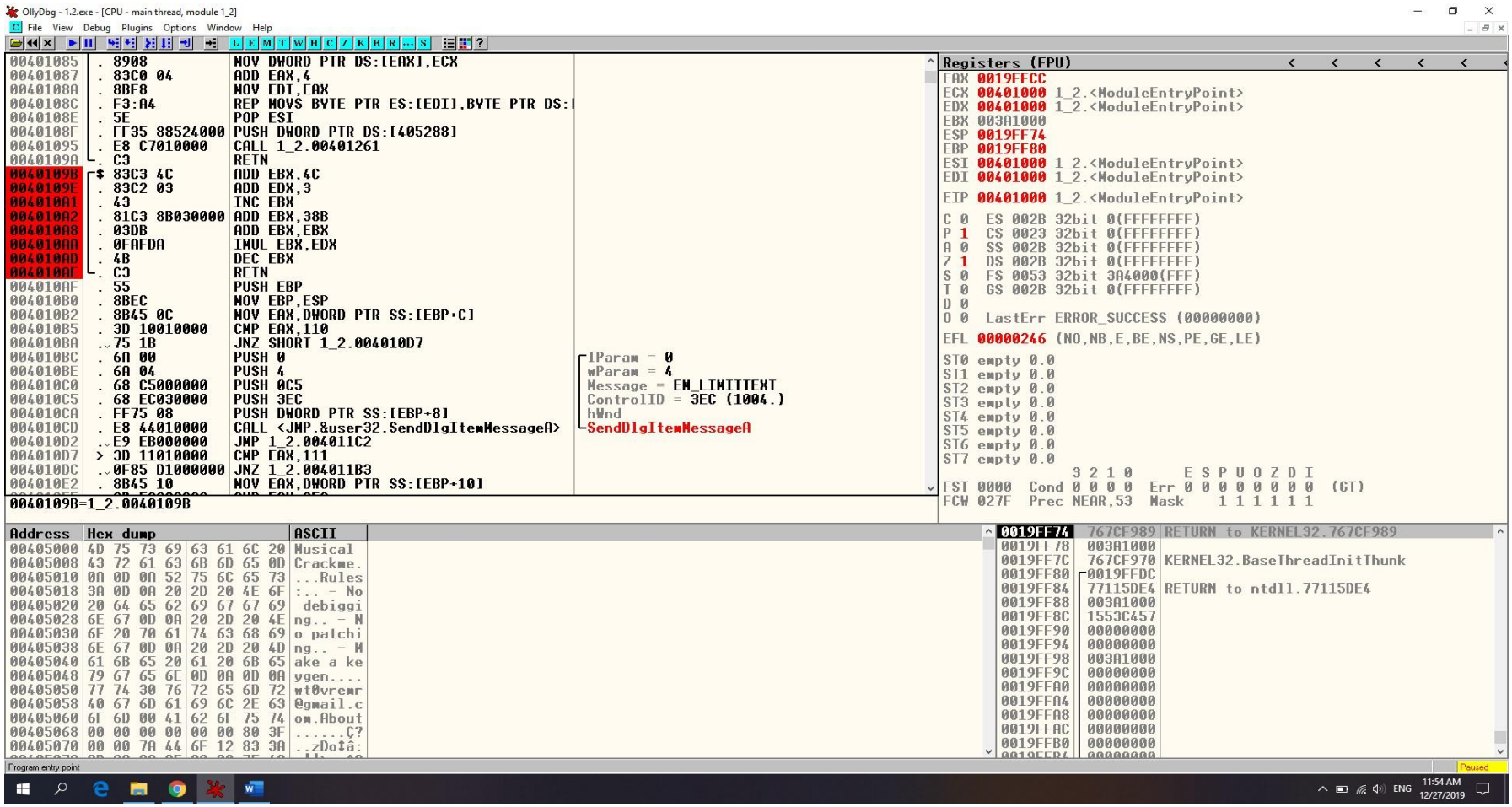
Mở ollydbg lên vào search tìm string thì ta chỉ thấy có chuỗi về about



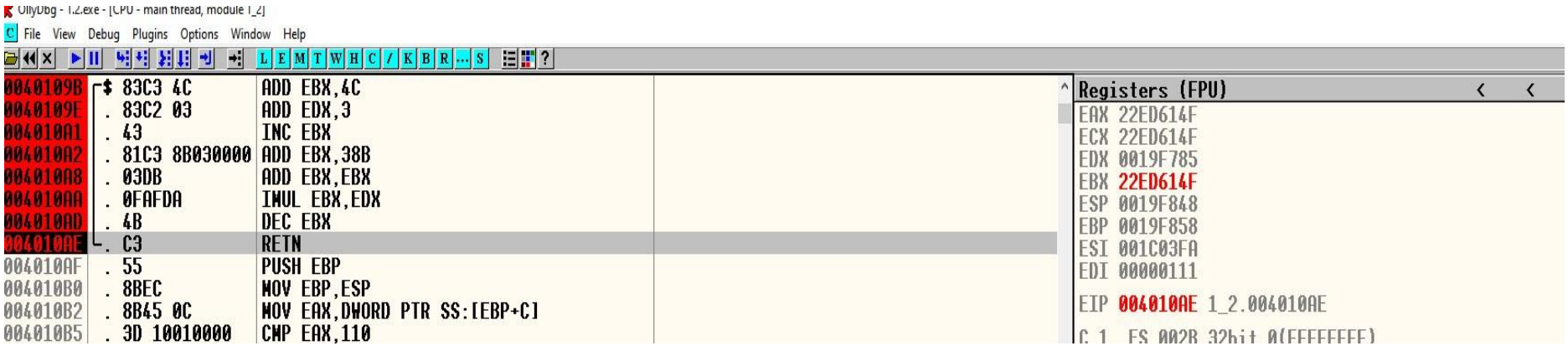
Sau đó ta vào search -> all intermodular calls thì phát hiện thấy có 2 dòng màu đỏ như trong hình là 2 dòng cho phép nhập ID và serial



Đi đến dòng đó và ta phát hiện sau khi nhập ID và serial xong ở dòng 00401149 Call 1_2.0040109B chương trình sẽ nhảy tới dòng 0040109B



Đi đến địa chỉ đó và ta thấy được những dòng màu đỏ trong hình là phát sinh ra serial từ ID ta đã nhập



Ta thử nhập ID là 1712 và debug những dòng trên để tìm serial và ta có được kết quả EBX là 22ED614F đổi ra hệ 10 là 585982287

ADD EBX, 4C -> EBX = EBX + 76

ADD EDX, 3 -> EDX = EDX + 3

INC EBX -> EBX++

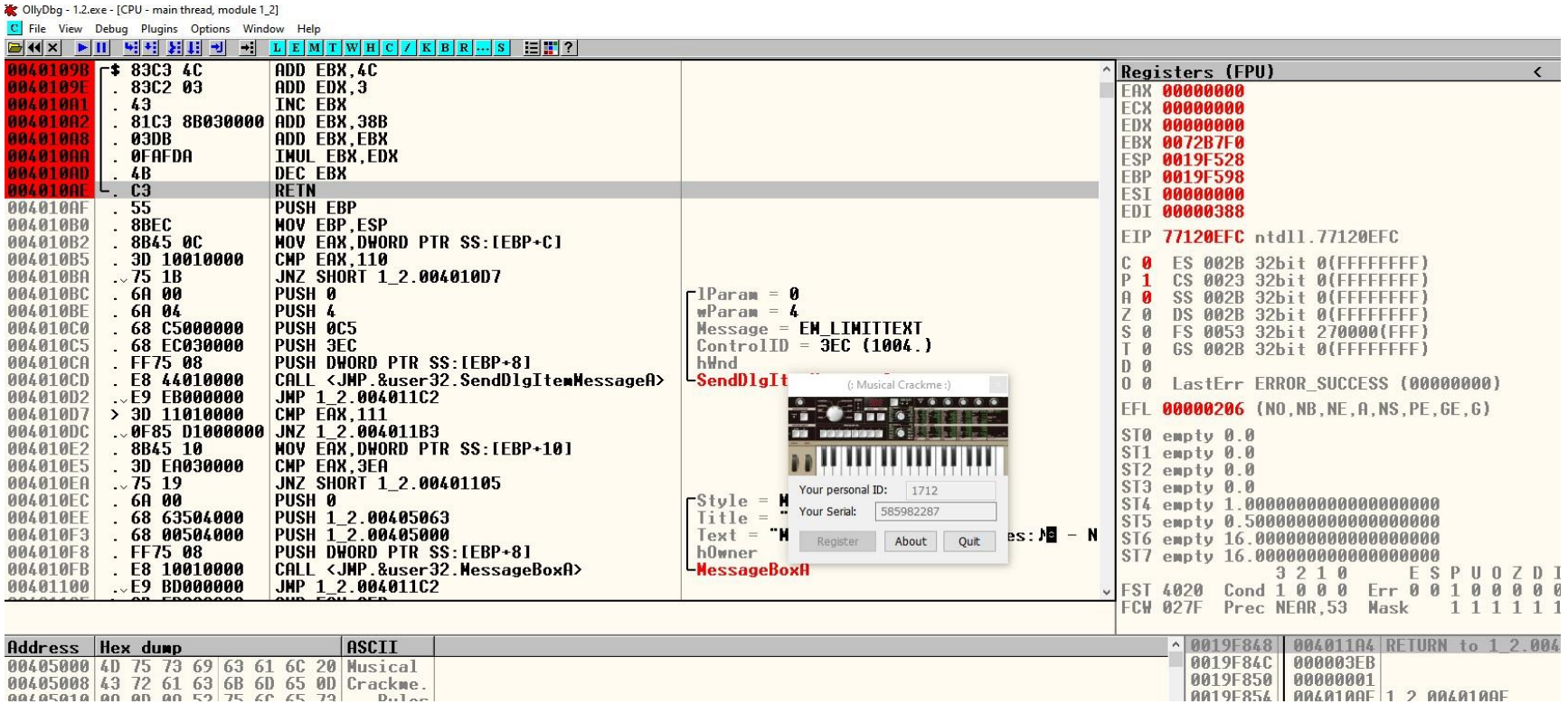
ADD EBX, 38B -> EBX = EBX + 907

ADD EBX, EBX -> EBX = EBX + EBX

IMUL EBX, EDX -> EBX = EBX * EDX

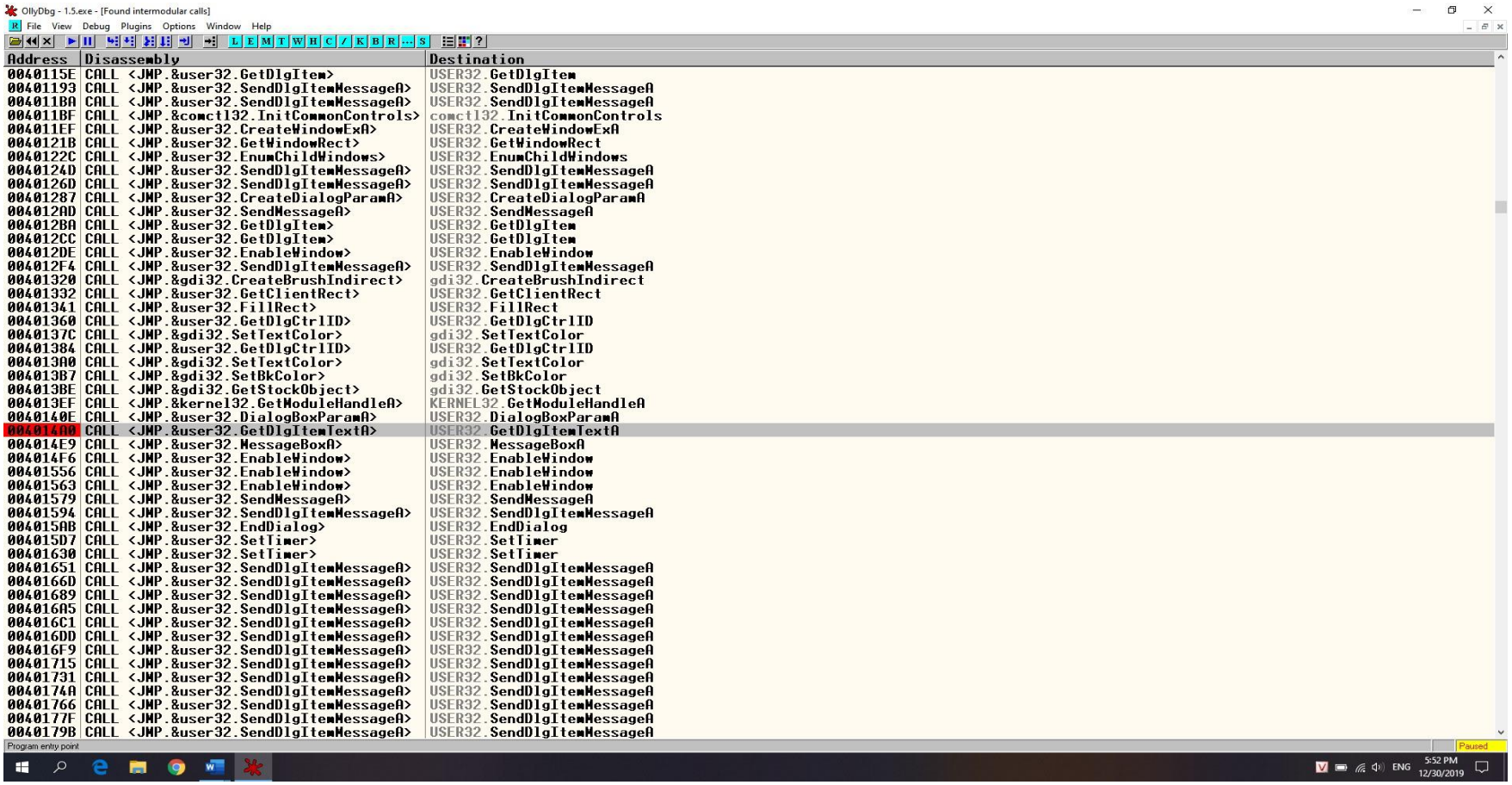
DEC EBX -> EBX--

Tóm lại ta có công thức tổng quát cho serial như sau: Serial = 3 * (2 * ID + 1968) - 1



Sau đó ta thử lại với ID và Serial ở trên thì thấy chương trình bị mờ và có tiếng nhạc phát lên -> ta đã nhập đúng ID và Serial

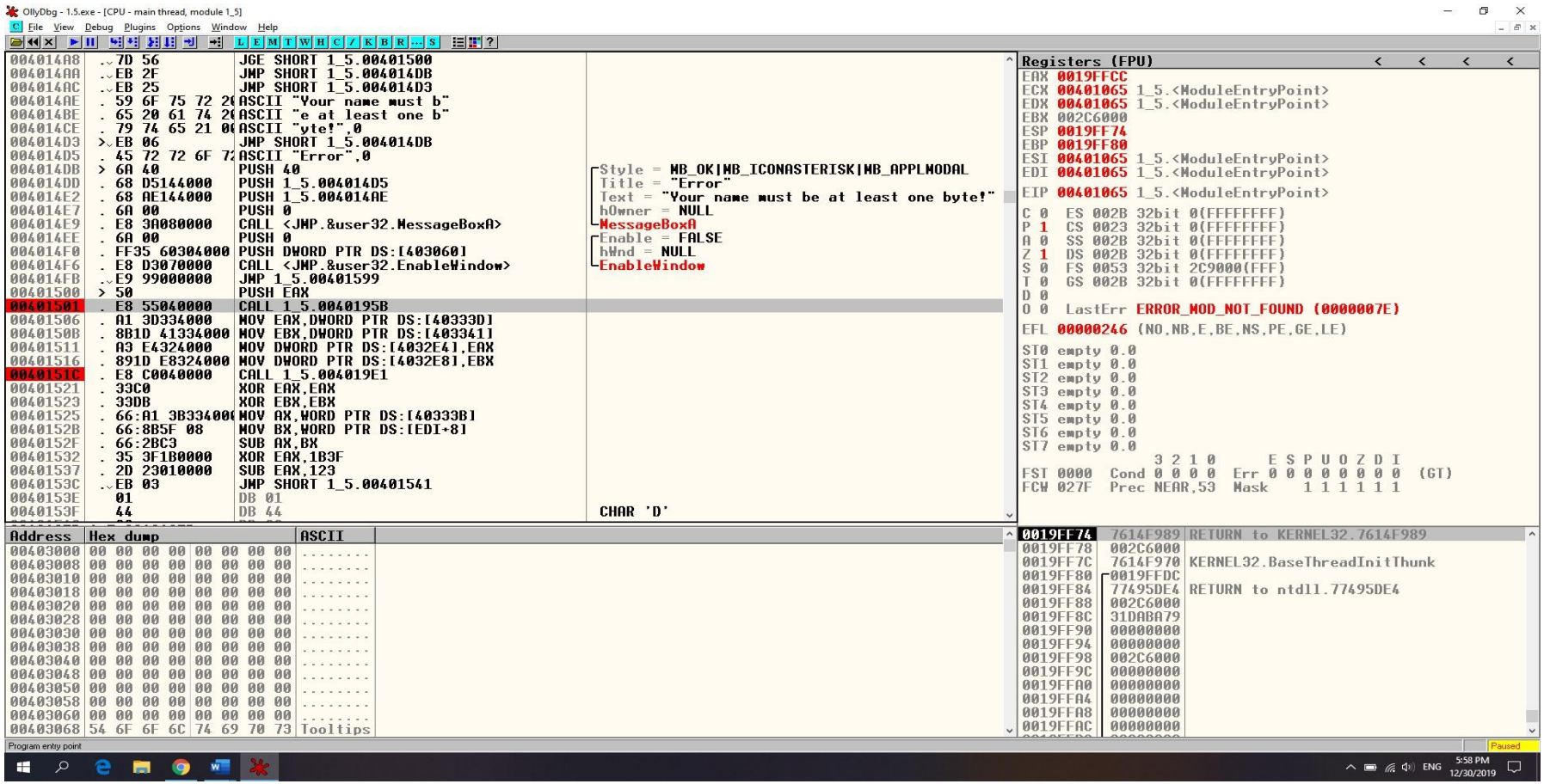
1.5



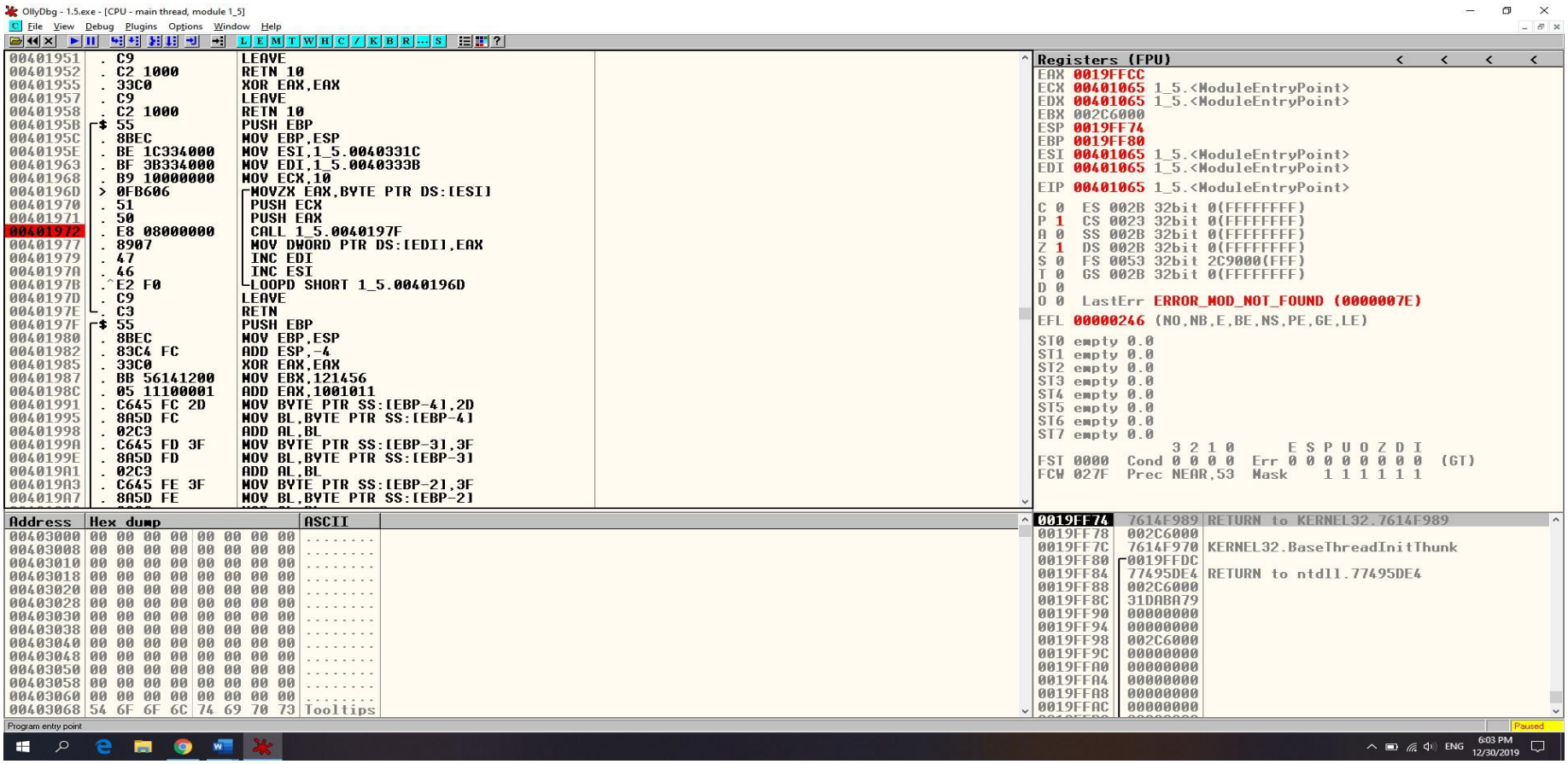
Đầu tiên mở chương trình lên vào search for -> all intermodular calls

Tại địa chỉ 4014A0 (ô màu đỏ như trong hình) là nơi để ta lấy key của chương trình, ta nhấn vào đó để đi đến địa chỉ đó.

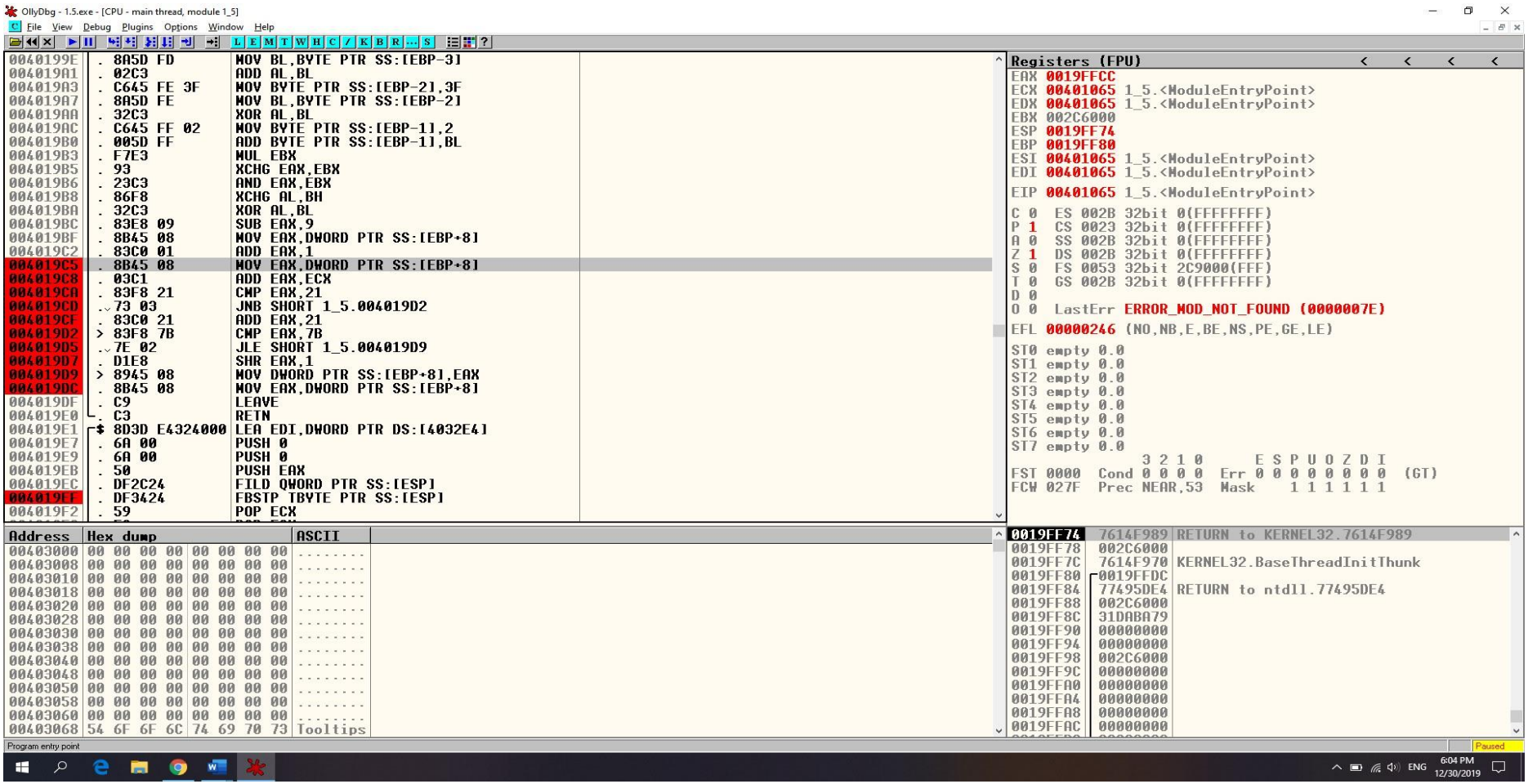
CMP EAX, 1 trên địa chỉ 004014A5 sẽ kiểm tra nếu có bất kỳ thứ gì đã được nhập vào hộp văn bản. Nếu không có gì được nhập thì sẽ nhận được hộp thông báo của ở địa chỉ 004014DB cho biết "Tên của bạn phải có ít nhất một byte".



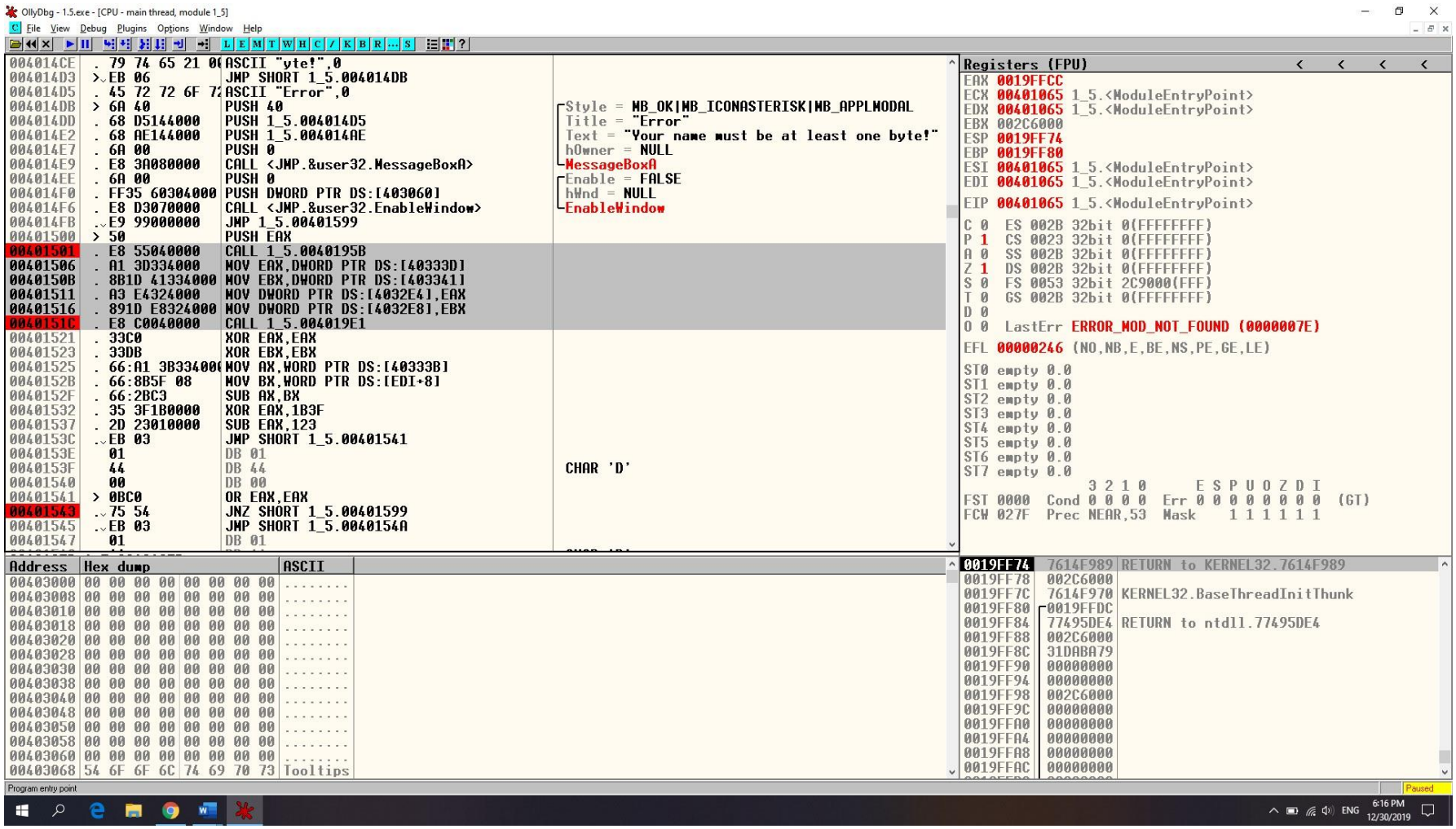
Tiếp đó nhấn f8 chương trình sẽ đi tới địa chỉ 00401501 (dòng màu đỏ đầu tiên trong hình), bắt đầu giai đoạn đoạn để lấy key



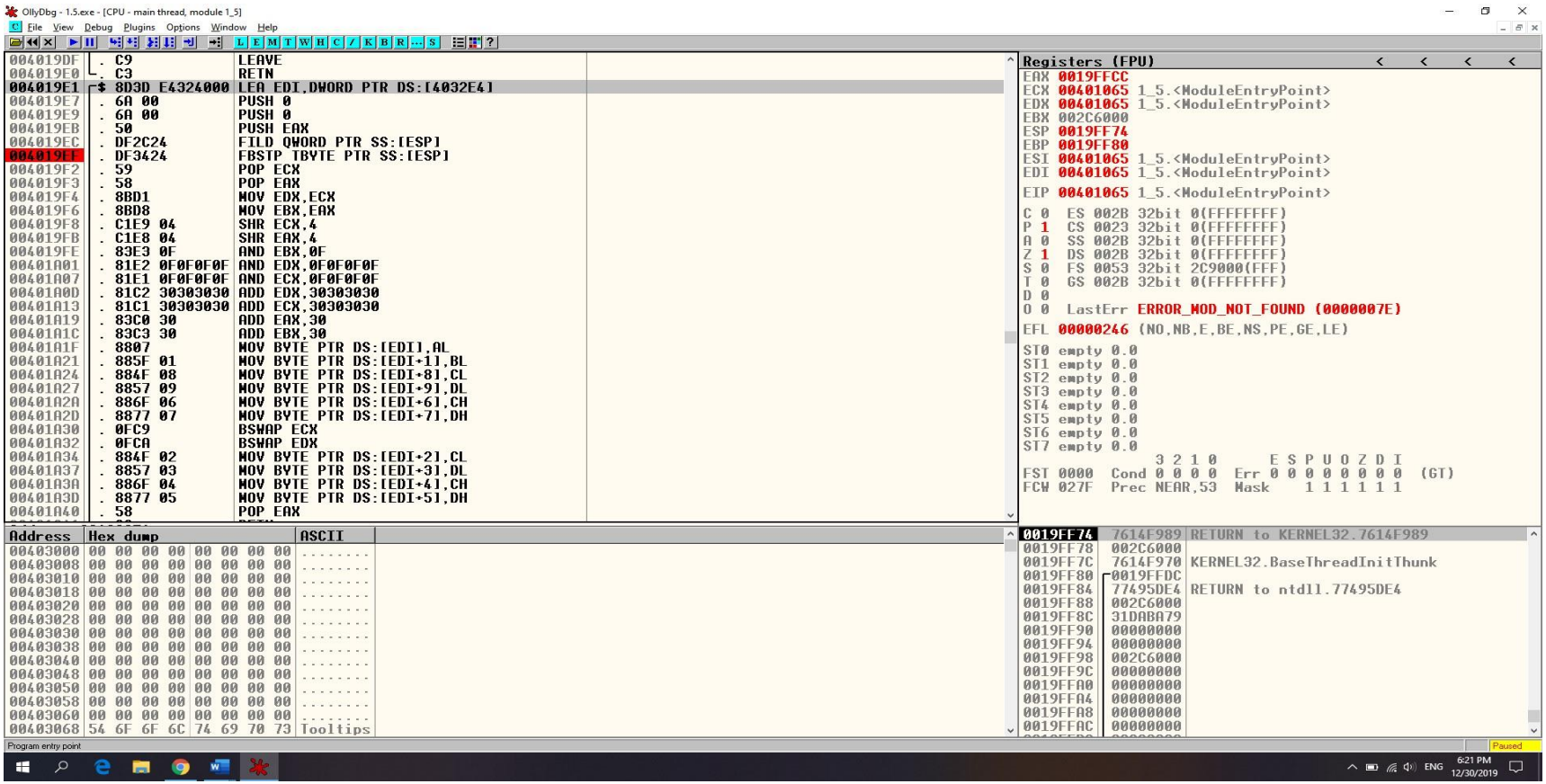
Tiếp tục nhấn f7 để di chuyển đến địa chỉ như trong hình



Tiếp tục di chuyển ta sẽ nhảy đến địa chỉ 0040197F, tuy nhiên từ địa chỉ 004019C5 đến 004019DC mới là đoạn mã hóa chính. Giá trị ascii (tương đương ở dạng hexa) của mỗi chữ cái/ kí tự của key vừa nhập được đặt ở EAX. Sau đó nó được cộng với giá trị ở ECX. Nếu EAX lớn hơn 21 sẽ nhảy tới địa chỉ 004019CC. Sau đó giá tri được so sánh với 7B thì sẽ thực hiện thao tác SHR EAX, 1. Giá trị này sau đó được lưu trữ. (SHL = shift logical right). Kí tự tiếp theo của chữ số đó được lấy ra và đặt ở EAX, ECX bị giảm và cứ thế lặp lại cho đến khi ECX = 0



Sau lần gọi cuối đó chúng ta sẽ quay lại 6 dòng như trong hình

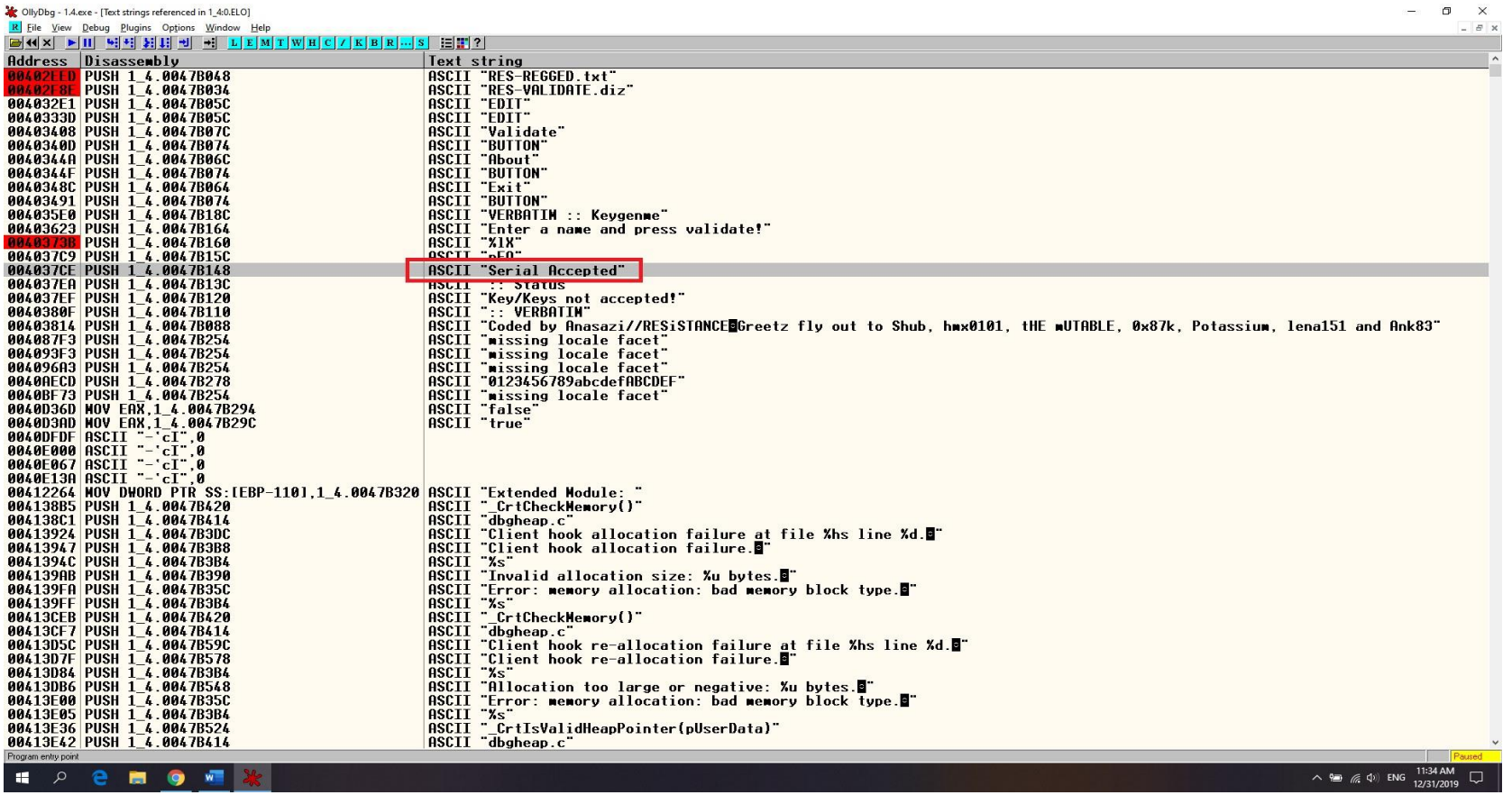


Sau đó nó sẽ nhảy tới địa chỉ 004019E1. Ở địa chỉ 004019EF sẽ chuyển đổi 4 bytes được lưu của key được mã hóa đến giá trị decimal. Phần mã còn lại định dạng thập phân, vd 123456789 thì giá trị decimal là 3938 được lưu trữ sau khi định dạng.

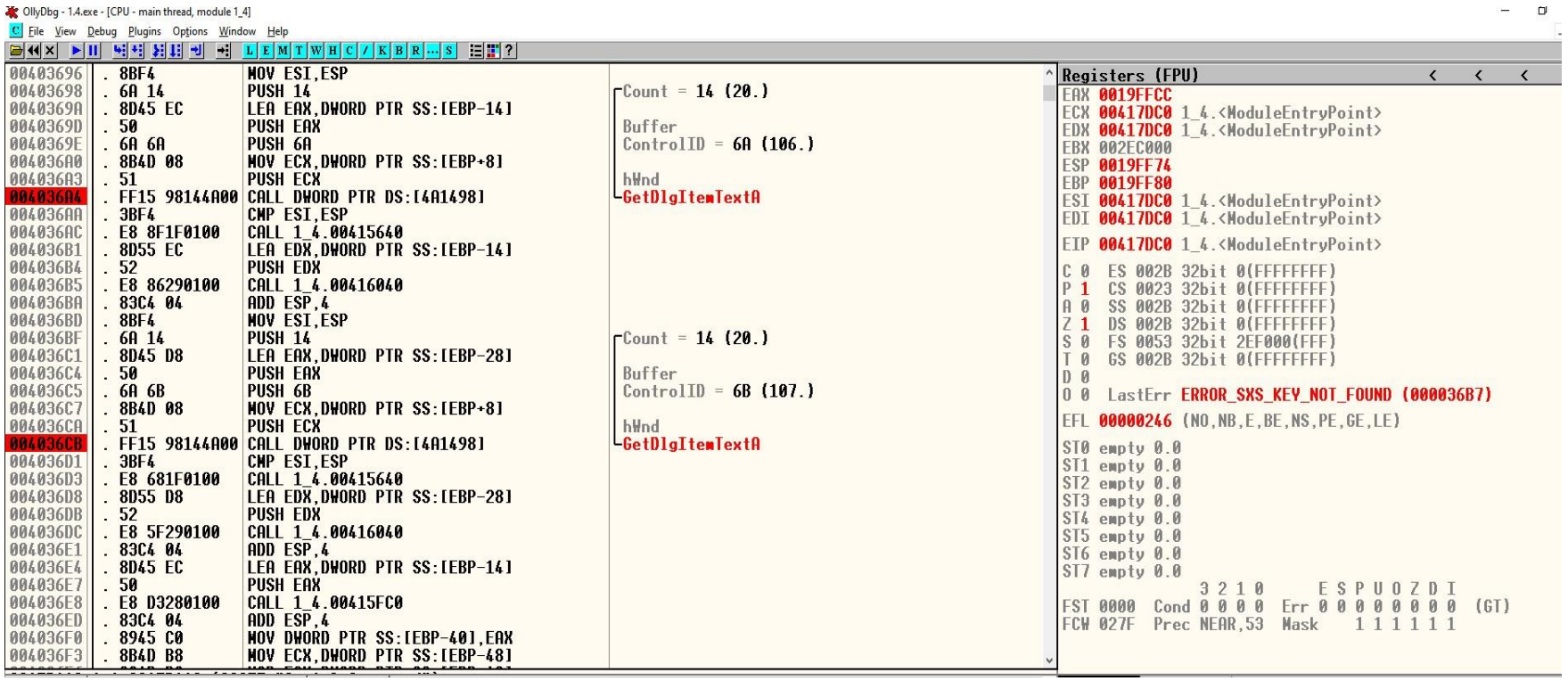
Hai bytes đầu tiên của key mã hóa được lưu ở địa chỉ 00401977 thì di chuyển đến EAX. Phần định dạng được chuyển đến EBX. Sau đó lấy EAX – EBX. Kết quả có được thực hiện phép xor với 1B3F và trừ cho 123. Nếu kết quả cuối cùng là 0 sẽ không nhảy tới địa chỉ 00401543 mà sẽ nhận được thông báo đăng ký thành công.

1.4

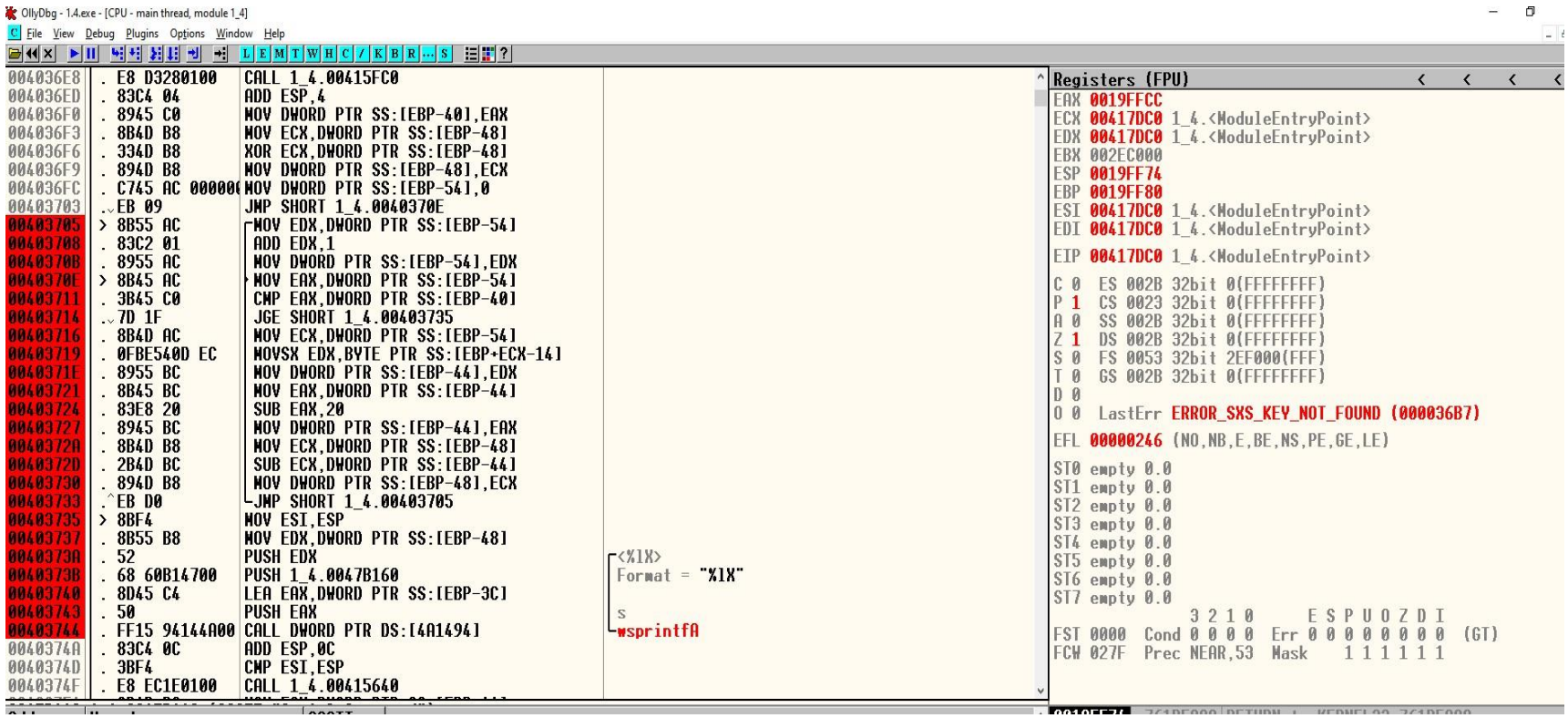
Đầu tiên mở chương trình lên, vào serach for -> all referenced text strings



Ta thấy ở địa chỉ 004037CE là chuỗi thông báo serial accepted, ta tiếp tục nhấn vào dòng đó để đi đến địa chỉ đó



Sau đó ta kéo lên trên thấy được tại hai dòng đỏ như trong hình là dòng cho người dùng nhập Name và Serial



Tiếp tục kéo xuống dưới ta thấy được từ địa chỉ 00403705 -> 00403733 là đoạn phát ra Serial từ Name ta đã nhập

004036D8 8D55 D8 LEA EDX,DWORD PTR SS:[EBP-28] → serial chuyển đến EDX

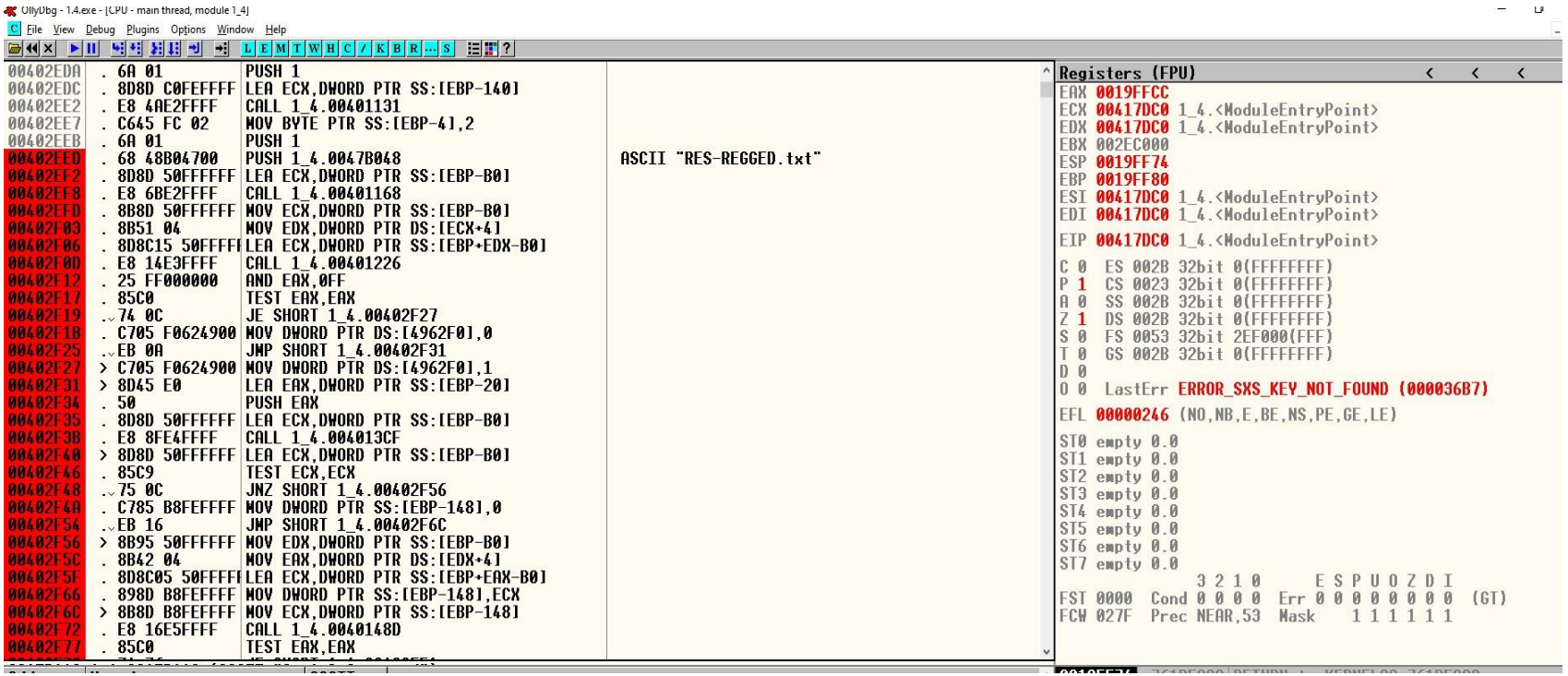
004036DB 52 PUSH EDX → lưu serial

004036E4 8D45 EC LEA EAX,DWORD PTR SS:[EBP-14] → đảo ngược Name đến EAX

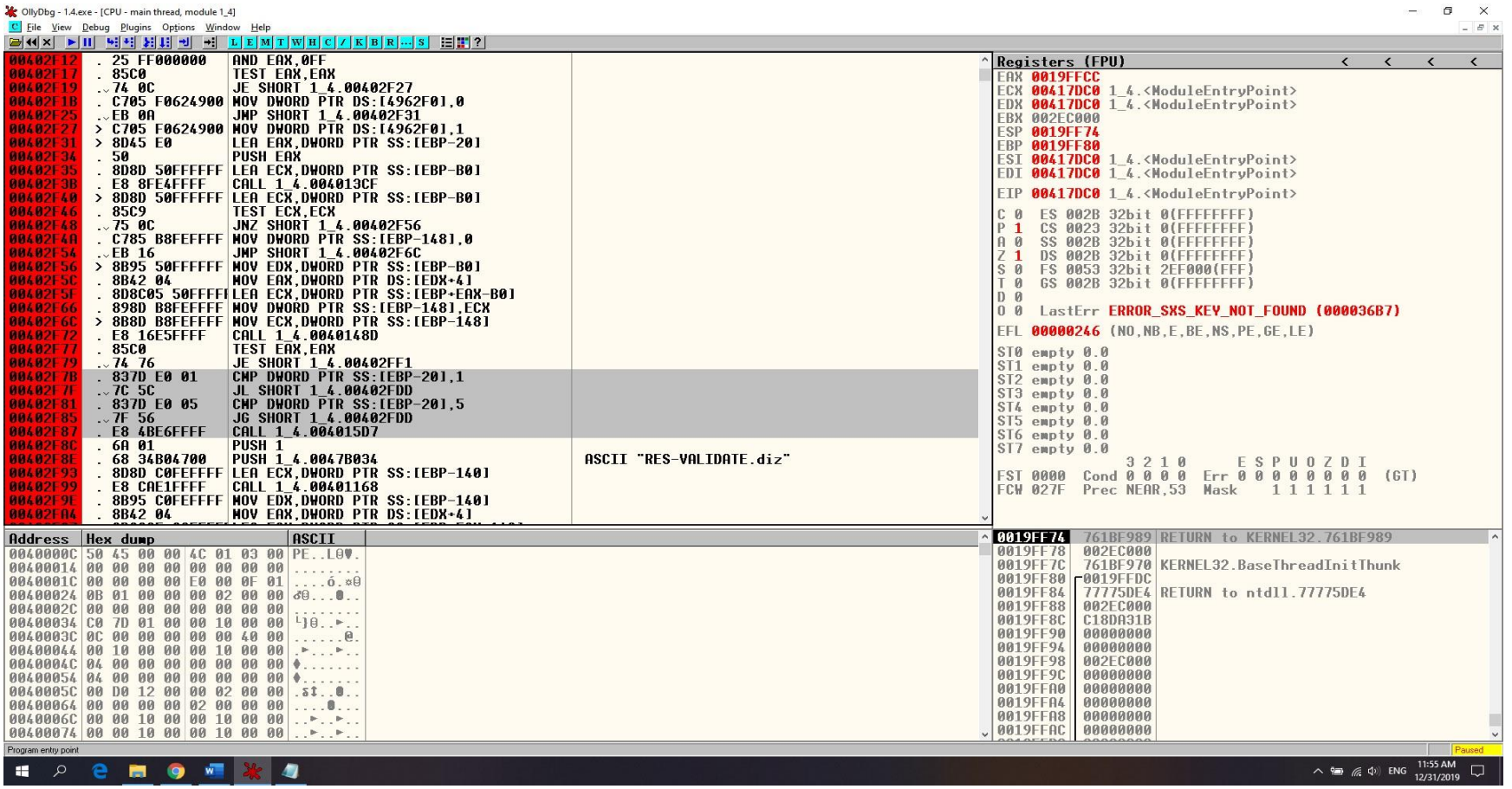
004036E7 50 PUSH EAX → lưu tên đảo ngược

004036F3 8B4D B8 MOV ECX,DWORD PTR SS:[EBP-48] → đảo ngược tên đến ECX

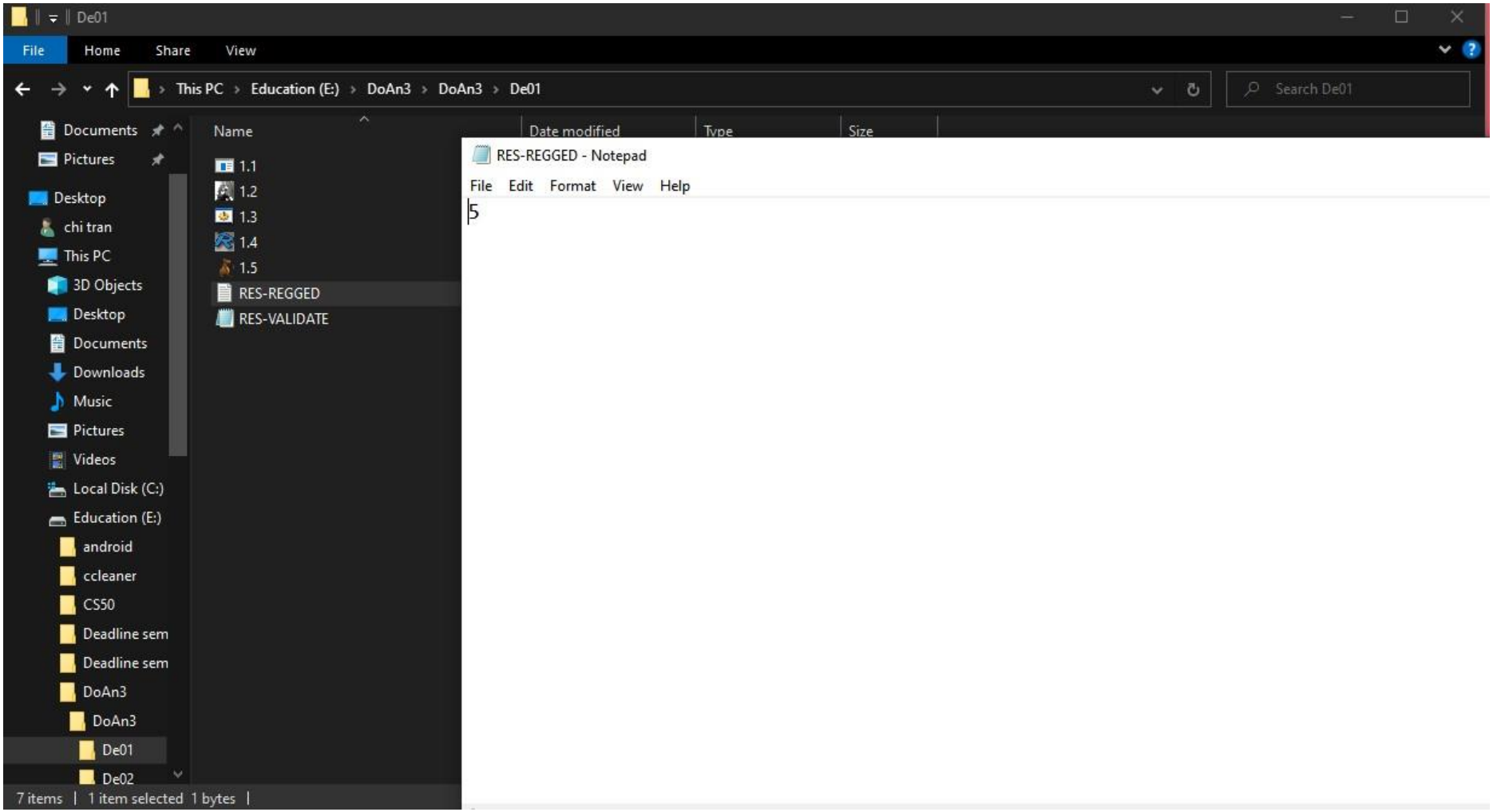
Về phần phát sinh key những dòng màu đỏ như trong hình có thể giải thích ngắn gọn như sau: ban đầu từ chuỗi nam người dùng nhập vào chương trình sẽ đảo ngược chuỗi Name, sau đó dùng vòng lặp với biến đếm là EDX để lấy ra từng ký tự của chuỗi Name đã bị đảo ngược gán vào EAX rồi trừ cho 20 ở hệ Hexa và lưu kết quả. Sau đó lấy ECX trừ cho kết quả đó. Sau khi chạy vòng lặp ta có được kết quả ở Hexa kiểu 8 bits, đảo ngược lại kết quả đó ta có được Serial, vd với Name nhập vào là “a” thì ta có được Serial tương ứng sau khi đảo ngược là: “FBFFFFFF”. Tuy nhiên đến đây tuy ta đã có được Name và Serial của chương trình những nếu nhập vào thì vẫn bị báo là sai.



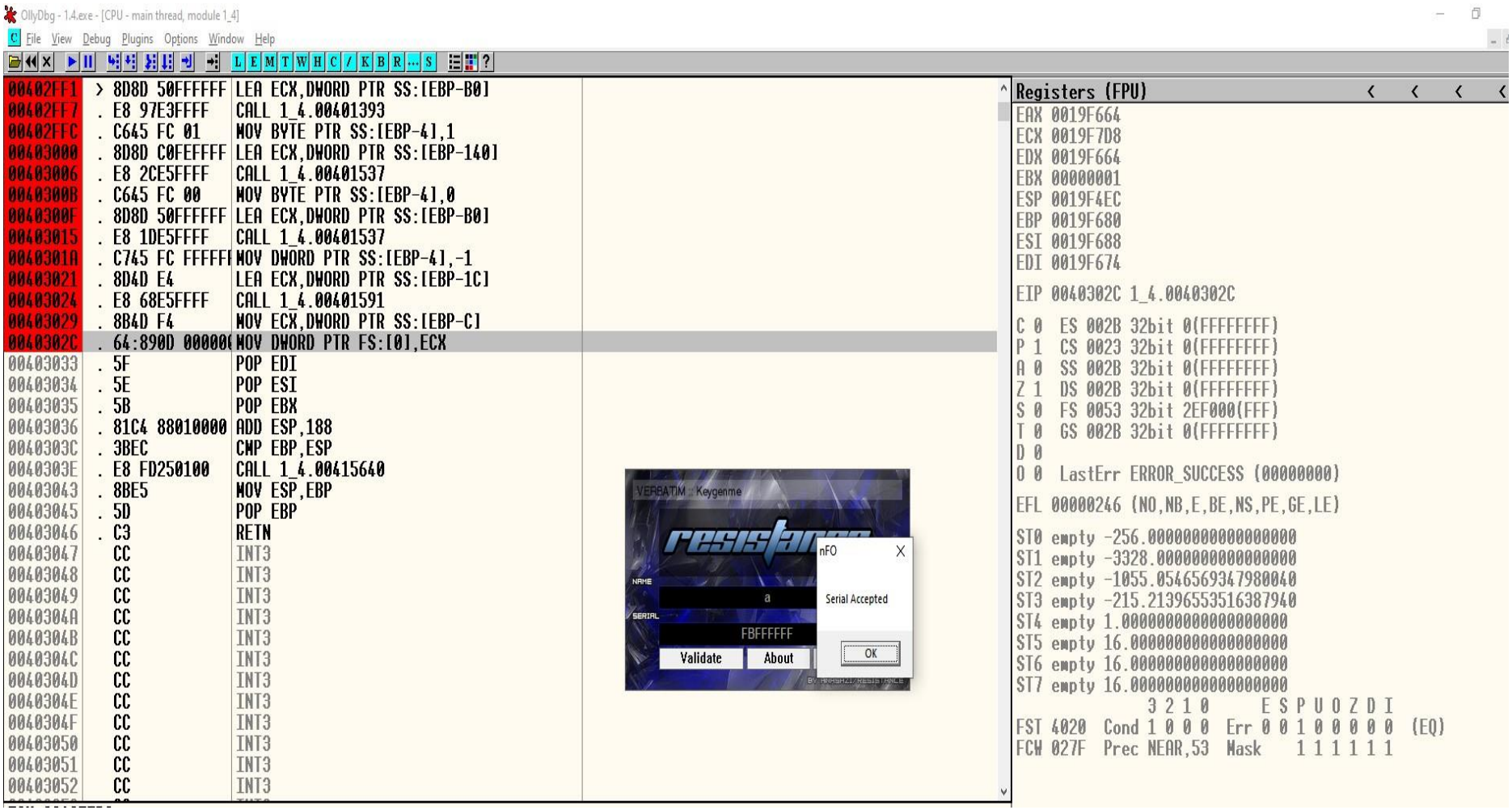
Đến đây ta kéo lên trên và phát hiện ở địa chỉ 00402EF2 chương trình sẽ kiểm tra file RES-REGGED.txt



Sau đó từ địa chỉ 00402F7F -> 00402F87 sẽ kiểm tra xem số trong file đó < 1 hoặc > 5, tiếp theo ở địa chỉ 00402F93 chương trình sẽ kiểm tra tiếp 1 file có tên là RES-VALIDATE.diz. Ta có nhận xét file RES-VALIDATE.diz chỉ cần được đặt chung trong folder với file crack còn lại ko có dữ liệu gì trong đó còn file RES-REGGED.txt sẽ có 1 con số trong khoảng từ 1→5



Ta cũng tạo 2 file như vậy và nhập vào thử số 5 trong file RES-REGGED.txt và debug chương trình lại 1 lần nữa với Name và Serial ta đã có được ở trên và kết quả là chương trình hiện thông báo thành công



II. Mức độ hoàn thành

- Đánh giá mức độ hoàn thành: 100%