

BÁO CÁO ĐỒ ÁN 3 – CRACKING

I. Thành viên nhóm

STT	Tên thành viên	MSSV
1	Nguyễn Bảo Long	18120201
2	Võ Thế Minh	18120211
3	Phạm Văn Minh Phương	18120227
4	Trà Anh Toàn	18120662
5	Mai Ngọc Tú	18120253

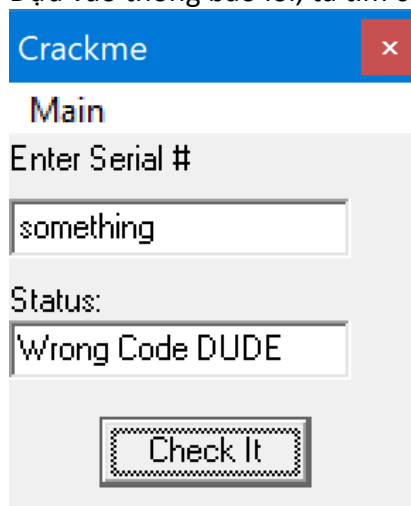
II. Phân công công việc và đánh giá mức độ hoàn thành

STT	Tên công việc	Yêu cầu	Người thực hiện	Đánh giá
1	Câu 3.1		Phạm Văn Minh Phương	100%
2	Câu 3.2	Cơ bản	Trà Anh Toàn	100%
		Nâng cao	Mai Ngọc Tú	100%
3	Câu 3.3	Cơ bản	Nguyễn Bảo Long	100%
		Nâng cao	Võ Thế Minh	100%
4	Viết báo cáo		Nguyễn Bảo Long	100%

III. Quá trình crack phần mềm

1. Bài 3.1

- Dựa vào thông báo lỗi, ta tìm chuỗi **Wrong Code DUDE**



- Vô tình, chúng ta thấy được dòng thông báo **Thanks you made it** – ám chỉ việc active thành công phần mềm

00420537	MOV EDX, 3_1.00420590	ASCII "Benadryl"
00420543	MOV EDX, 3_1.004205A4	ASCII "Wrong Code DUDE"
00420555	MOV EDX, 3_1.004205BC	ASCII "Thanks you made it"
00420590	ASCII "Benadryl", 0	
004205A4	ASCII "Wrong Code DUDE", 0	
004205BC	ASCII "Thanks you made it", 0	

- Sau đó chúng ta đi đến đoạn code gọi đến thông báo **Thanks you made it** và **Wrong Code DUDE**

0042D510	. 55	PUSH EBP	
0042D511	. 8BEC	MOV EBP,ESP	
0042D513	. 6A 00	PUSH 0	
0042D515	. 53	PUSH EBX	
0042D516	. 8BD8	MOV EBX,EAX	
0042D518	. 33C0	XOR EAX,EAX	
0042D51A	. 55	PUSH EBP	
0042D51B	. 68 7BD54200	PUSH 3_1.0042D57B	
0042D520	. 64:FF30	PUSH DWORD PTR FS:[EAX]	
0042D523	. 64:8920	MOV DWORD PTR FS:[EAX],ESP	
0042D526	. 8D55 FC	LEA EDX,DWORD PTR SS:[EBP-4]	
0042D529	. 8B83 DC010000	MOV EAX,DWORD PTR DS:[EBX+1DC]	
0042D52F	. E8 54CCFEFF	CALL 3_1.0041A1B8	
0042D534	. 8B45 FC	MOV EAX,DWORD PTR SS:[EBP-4]	
0042D537	. BA 90D54200	MOV EDX,3_1.0042D590	ASCII "Benadryl"
0042D53C	. E8 8F63FDFF	CALL 3_1.004038D0	
0042D541	. 74 12	JE SHORT 3_1.0042D555	
0042D543	. BA A4D54200	MOV EDX,3_1.0042D5A4	ASCII "Wrong Code DUDE"
0042D548	. 8B83 E8010000	MOV EAX,DWORD PTR DS:[EBX+1E8]	
0042D54E	. E8 65CCFEFF	CALL 3_1.0041A1B8	
0042D553	. EB 10	JMP SHORT 3_1.0042D565	
0042D555	. BA BCD54200	MOV EDX,3_1.0042D5BC	ASCII "Thanks you made it"
0042D55A	. 8B83 E8010000	MOV EAX,DWORD PTR DS:[EBX+1E8]	
0042D560	. E8 53CCFEFF	CALL 3_1.0041A1B8	
0042D565	. 33C0	XOR EAX,EAX	
0042D567	. 5A	POP EDX	
0042D568	. 59	POP ECX	
0042D569	. 59	POP ECX	
0042D56A	. 64:8910	MOV DWORD PTR FS:[EAX],EDX	
0042D56D	. 68 82D54200	PUSH 3_1.0042D582	
0042D572	. 8D45 FC	LEA EAX,DWORD PTR SS:[EBP-4]	
0042D575	. E8 CA5FFDFF	CALL 3_1.00403544	
0042D57A	. C3	RETN	

- Đặt breakpoint ở dòng **0042D510 PUSH EBP**. Ta nhập một giá trị bất kỳ vào khung Enter Serial rồi chạy debug từng dòng.

Crackme

×

Main

Enter Serial #

123

Status:

Check It

- Để ý rằng, khi chạy qua dòng **MOV EDX,3_1.0042D590** thì thanh ghi **EAX** mang giá trị vừa nhập vào **123**

MOV EAX,DWORD PTR SS:[EBP-4]	
MOV EDX,3_1.0042D590	ASCII "Benadryl"
CALL 3_1.004038D0	
JE SHORT 3_1.0042D555	
MOV EDX,3_1.0042D5A4	ASCII "Wrong Code D
MOV EAX,DWORD PTR DS:[EBX+1E8]	

Registers (FPU)		<	<
EAX	0225A3B8	ASCII "123"	
ECX	0D7B7D57		
EDX	00000000		
EBX	02251A4C		
ESP	00105400		

- Tiếp tục debug từng dòng lệnh, ta đến được đoạn code chương trình so sánh 2 thanh ghi **EAX** chứa giá trị **123** với thanh ghi **EDX** chứa giá trị **Benadryl**. Tới đây, ta hoàn toàn có quyền nghi ngờ rằng **Benadryl** chính là giá trị cần tìm.

```
MOV EDI,EDX
CMP EAX,EDX
JE 3_1.0040396E
TEST ESI,ESI
```

Registers (FPU)		
EAX	0225A3B8	ASCII "123"
ECX	0D7B7D57	
EDX	0042D590	ASCII "Benadryl"
EBX	02251A4C	

- Thử nhập **Benadryl** vào khung Enter Serial, ta được kết quả sau

2. Bài 3.2

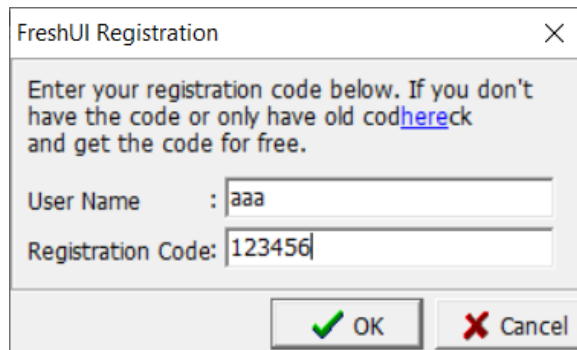
- Tìm GOOD BOY, BAD BOY: Thử nhập bất kỳ 1 Username và 1 Registration Code, ta không thấy thông báo. Tiến hành đọc code hợp ngữ và tìm kiếm với những string khả nghi như "Success", "Successfully",... ta tìm được:

00490CEB	. 64:8910	MOV DWORD PTR FS:[EAX],EDX	
00490CEE	. 68 120F4900	PUSH 3_2.00490F12	
00490CF3	> 8B45 FC	MOV EAX,DWORD PTR SS:[EBP-4]	
00490CF6	. E8 C1DFFFF	CALL 3_2.00490ABC	
00490CFB	. 833D 80AC5200	CMP DWORD PTR DS:[52AC80],0	
00490D02	. 74 0F	JE SHORT 3_2.00490D13	
00490D04	. A1 48985200	MOV EAX,DWORD PTR DS:[529848]	
00490D09	. 8B00	MOV EAX,DWORD PTR DS:[EAX]	
00490D0B	. 8B10	MOV EDX,DWORD PTR DS:[EAX]	
00490D0D	. FF92 F8000000	CALL DWORD PTR DS:[EDX+F8]	
00490D13	> 833D 78AC5200	CMP DWORD PTR DS:[52AC78],1	
00490D1A	. 1BC0	SBB EAX,EAX	
00490D1C	. 40	INC EAX	
00490D1D	. 3C 01	CMP AL,1	
00490D1F	. 74 12	JE SHORT 3_2.00490D33	
00490D21	. 833D 7CAC5200	CMP DWORD PTR DS:[52AC7C],1	
00490D28	. 1BC0	SBB EAX,EAX	
00490D2A	. 40	INC EAX	
00490D2B	. 3C 01	CMP AL,1	
00490D2D	. 0F85 B4010000	JNZ 3_2.00490EE7	
00490D33	> B8 6C0F4900	MOV EAX,3_2.00490F6C	ASCII "FreshUI has been registered successfully."
00490D38	. E8 0767FBFF	CALL 3_2.00447444	
00490D3D	. B2 01	MOV DL,1	
00490D3F	. A1 E0464200	MOV EAX,DWORD PTR DS:[4246E0]	
00490D44	. E8 973AF9FF	CALL 3_2.004247E0	
00490D49	. A3 84AC5200	MOV DWORD PTR DS:[52AC84],EAX	
00490D4E	. BA 02000000	MOV EDI,00000002	
00490D53	. A1 84AC5200	MOV EAX,DWORD PTR DS:[52AC84]	
00490D58	. E8 23BF99FF	CALL 3_2.00424800	
00490D5D	. B1 01	MOV CL,1	
00490D5F	. BA A00F4900	MOV EDI,3_2.00490FA0	ASCII "\\Software\FreshDevices\FreshUI"
00490D64	. A1 84AC5200	MOV EAX,DWORD PTR DS:[52AC84]	
00490D69	. E8 563CF9FF	CALL 3_2.004249C4	
00490D6E	. 8B00 80AC5200	MOV ECX,DWORD PTR DS:[52AC88]	
00490D74	. BA C30F4900	MOV EDI,3_2.00490FC3	ASCII "Owner"
00490D79	. A1 84AC5200	MOV EAX,DWORD PTR DS:[52AC84]	
00490D7E	. E8 4D43F9FF	CALL 3_2.004250D0	
00490D83	. 8B0D 8CAC5200	MOV ECX,DWORD PTR DS:[52AC8C]	
00490D89	. BA D30F4900	MOV EDI,3_2.00490FD8	ASCII "RegCode"
00490D8E	. A1 84AC5200	MOV EAX,DWORD PTR DS:[52AC84]	
00490D93	. E8 3843F9FF	CALL 3_2.004250D0	
00490D98	. 833D 7CAC5200	CMP DWORD PTR DS:[52AC7C],0	
00490D9F	. 74 09	JE SHORT 3_2.00490DAA	
00490DA1	. C745 F8 7FD5B	MOV DWORD PTR SS:[EBP-8],2B0D57F	
00490DA8	. EB 07	JMP SHORT 3_2.00490DB1	
00490DAA	. C745 F8 92600	MOV DWORD PTR SS:[EBP-8],20A6092	
00490DB1	. 6A 04	PUSH 4	Arg1 = 00000004
00490DB3	. 8D4D F8	LEA ECX,DWORD PTR SS:[EBP-8]	
00490DB6	. BA E80F4900	MOV EDI,3_2.00490FE8	ASCII "Registered"
00490DBB	. A1 84AC5200	MOV EAX,DWORD PTR DS:[52AC84]	

- Nhìn vào địa chỉ của dòng **FreshUI has been registered successfully**, ta nhận thấy ở phía trên có một lệnh nhảy chạy đến thông báo này. Đặt breakpoint tại đó.

00490CF3	> 8B45 FC	MOV EAX,DWORD PTR SS:[EBP-4]	
00490CF6	. E8 C1DFFFF	CALL 3_2.00490ABC	
00490CFB	. 833D 80AC5200	CMP DWORD PTR DS:[52AC80],0	
00490D02	. 74 0F	JE SHORT 3_2.00490D13	
00490D04	. A1 48985200	MOV EAX,DWORD PTR DS:[529848]	
00490D09	. 8B00	MOV EAX,DWORD PTR DS:[EAX]	
00490D0B	. 8B10	MOV EDX,DWORD PTR DS:[EAX]	
00490D0D	. FF92 F8000000	CALL DWORD PTR DS:[EDX+F8]	
00490D13	> 833D 78AC5200	CMP DWORD PTR DS:[52AC78],1	
00490D1A	. 1BC0	SBB EAX,EAX	
00490D1C	. 40	INC EAX	
00490D1D	. 3C 01	CMP AL,1	
00490D1F	. 74 12	JE SHORT 3_2.00490D33	
00490D21	. 833D 7CAC5200	CMP DWORD PTR DS:[52AC7C],1	
00490D28	. 1BC0	SBB EAX,EAX	
00490D2A	. 40	INC EAX	
00490D2B	. 3C 01	CMP AL,1	
00490D2D	. 0F85 B4010000	JNZ 3_2.00490EE7	
00490D33	> B8 6C0F4900	MOV EAX,3_2.00490F6C	ASCII "FreshUI has been registered succe
00490D38	. E8 0767FBFF	CALL 3_2.00447444	
00490D3D	. B2 01	MOV DL,1	
00490D3F	. A1 E0464200	MOV EAX,DWORD PTR DS:[4246E0]	

- Trên dòng **JE SHORT 3_2.00490D33** có lệnh **CMP AL, 1**, nghĩa là sau khi thực hiện hàm nào đó, nếu giá trị **AL = 1** thì nhảy tới thông báo **FreshUI has been registered successfully**.
- Tiếp tục debug trên từng dòng, ta nhận thấy, để thực thi lệnh **JE SHORT 3_2.00490D33**, ta cần phải thực hiện lệnh **JE SHORT 3_2.00490D13**. Như vậy, phải cho giá trị ở ô nhớ **52AC80** bằng 0 (**[52AC80] = 0**)
- Ta xét lệnh **CALL 3_2.00490ABC** – hàm gần nhất để nhảy tới lệnh thông báo nhập key đúng. Đặt breakpoint ở dòng này và nhấn F9 để nhảy đến đó. Sau đó ngưng lại.
- Nhập ngẫu nhiên 1 giá trị User Name và Registration Code.



- Chạy từng dòng lệnh để chương trình chạy đến đoạn lệnh mà **CALL 3_2.00490ABC** gọi tới

00490ADB	. 33C0	XOR EAX,EAX
00490ADD	. A3 78AC5200	MOV DWORD PTR DS:[52AC78],EAX
00490AE2	. 33C0	XOR EAX,EAX
00490AE4	. A3 7CAC5200	MOV DWORD PTR DS:[52AC7C],EAX
00490AE9	. 33C0	XOR EAX,EAX
00490AEB	. A3 80AC5200	MOV DWORD PTR DS:[52AC80],EAX

- Ta thấy những ô nhớ **[52AC78]**, **[52AC7C]** và **[52AC80]** đều là những ô nhớ mang giá trị quyết định cho điều kiện lệnh nhảy để nhảy tới GOODBOY như những hình trên. Gán giá trị những ô nhớ đó bằng **EAX**, mà **XOR EAX, EAX** nghĩa là gán **EAX = 0**

→ Có thể đây là ô nhớ chứa biến Boolean dùng kiểm tra xem key nhập vào đúng hay sai.

- Để kiểm tra giả thuyết, ta chọn dòng có ô nhớ **[52AC78]** → nhấn chuột phải → Find references to → Address constant. Một hộp thoại với những dòng lệnh có gọi ô nhớ **[52AC78]** sẽ xuất hiện.

Address	Disassembly	Comment
00490ADD	MOV DWORD PTR DS:[52AC78],EAX	(Initial CPU selection)
00490BCC	MOV DWORD PTR DS:[52AC78],-1	DS:[0052AC78]=00000000
00490D13	CMP DWORD PTR DS:[52AC78],1	DS:[0052AC78]=00000000

- Giá trị của ô nhớ này chỉ thay đổi bởi 3 lệnh, nhấn đúp vào từng lệnh để xem vị trí thì thấy nó đều nằm trong hàm chúng ta đang xét. Giá trị của ô nhớ được khởi tạo lần đầu khi gán với **EAX**, và chỉ thay đổi xoay quanh giá trị 0 và -1

→ Giả thuyết đặt ra có thể đúng (nếu key đúng thì gán 0, key sai gán -1).

- Làm tương tự với hai ô nhớ còn lại cũng cho kết quả tương tự. Tại sao lại cần tới 3 biến kiểm tra key?

→ Có thể có 3 loại key.

- Tiếp tục chạy từng dòng lệnh, ta bắt gặp 3 vòng lặp như sau

00490B46	.v7C 3C	JL SHORT 3_2.00490B84
00490B48	. 43	INC EBX
00490B49	. 33F6	MOV ESI,ESI
00490B4B	> 8D4D F0	LEA ECX,DWORD PTR SS:[EBP-10]
00490B4E	. 0FBFD6	MOVSX EDX,SI
00490B51	. 8B45 FC	MOV EAX,DWORD PTR SS:[EBP-4]
00490B54	. 8B80 64030000	MOV EAX,DWORD PTR DS:[EAX+364]
00490B5A	. 8B80 38020000	MOV EAX,DWORD PTR DS:[EAX+238]
00490B60	. 8B38	MOV EDI,DWORD PTR DS:[EAX]
00490B62	. FF57 0C	CALL DWORD PTR DS:[EDI+C]
00490B65	. 8B55 F0	MOV EDX,DWORD PTR SS:[EBP-10]
00490B68	. A1 8CAC5200	MOV EAX,DWORD PTR DS:[52AC8C]
00490B6D	. E8 663AF7FF	CALL 3_2.004045D8
00490B72	.v75 0A	JNZ SHORT 3_2.00490B7E
00490B74	. C705 80AC5200	MOV DWORD PTR DS:[52AC80],-1
00490B7E	> 46	INC ESI
00490B7F	. 66:FFCB	DEC BX
00490B82	.^75 C7	JNZ SHORT 3_2.00490B4B
00490B84	> 8B45 FC	MOV EAX,DWORD PTR SS:[EBP-4]
00490B87	. 8B80 5C030000	MOV EAX,DWORD PTR DS:[EAX+35C]
00490B8D	. 8B80 38020000	MOV EAX,DWORD PTR DS:[EAX+238]
00490B93	. 8B10	MOV EDX,DWORD PTR DS:[EAX]
00490B95	. FF52 14	CALL DWORD PTR DS:[EDX+14]
00490B98	. 8B08	MOV EBX,EAX
00490B9A	. 4B	DEC EBX
00490B9B	. 66:85DB	TEST BX,BX
00490B9E	.v7C 3C	JL SHORT 3_2.00490BDC
00490BA0	. 43	INC EBX
00490BA1	. 33F6	MOV ESI,ESI
00490BA3	> 8D4D EC	LEA ECX,DWORD PTR SS:[EBP-14]
00490BA6	. 0FBFD6	MOVSX EDX,SI
00490BA9	. 8B45 FC	MOV EAX,DWORD PTR SS:[EBP-4]
00490BAC	. 8B80 5C030000	MOV EAX,DWORD PTR DS:[EAX+35C]
00490BB2	. 8B80 38020000	MOV EAX,DWORD PTR DS:[EAX+238]
00490BB8	. 8B38	MOV EDI,DWORD PTR DS:[EAX]
00490BBA	. FF57 0C	CALL DWORD PTR DS:[EDI+C]
00490BBD	. 8B55 EC	MOV EDX,DWORD PTR SS:[EBP-14]
00490BC0	. A1 8CAC5200	MOV EAX,DWORD PTR DS:[52AC8C]
00490BC5	. E8 0E3AF7FF	CALL 3_2.004045D8
00490BCA	.v75 0A	JNZ SHORT 3_2.00490BD6
00490BCC	. C705 78AC5200	MOV DWORD PTR DS:[52AC78],-1
00490BD6	> 46	INC ESI
00490BD7	. 66:FFCB	DEC BX
00490BDA	.^75 C7	JNZ SHORT 3_2.00490BA3
00490BDC	> 8B45 FC	MOV EAX,DWORD PTR SS:[EBP-4]
00490BDF	. 8B80 60030000	MOV EAX,DWORD PTR DS:[EAX+360]
00490BE5	. 8B80 38020000	MOV EAX,DWORD PTR DS:[EAX+238]
00490BED	. 8B10	MOV EDX,DWORD PTR DS:[EAX]
00490BF0	. FF52 14	CALL DWORD PTR DS:[EDX+14]
00490BF2	. 8B08	MOV EBX,EAX
00490BF3	. 4B	DEC EBX
00490BF6	. 66:85DB	TEST BX,BX
00490BF8	.v7C 3C	JL SHORT 3_2.00490C34
00490BF9	. 43	INC EBX
00490BF9	. 33F6	MOV ESI,ESI
00490BF8	> 8D4D E8	LEA ECX,DWORD PTR SS:[EBP-18]
00490BFE	. 0FBFD6	MOVSX EDX,SI
00490C01	. 8B45 FC	MOV EAX,DWORD PTR SS:[EBP-4]
00490C04	. 8B80 60030000	MOV EAX,DWORD PTR DS:[EAX+360]
00490C0A	. 8B80 38020000	MOV EAX,DWORD PTR DS:[EAX+238]
00490C10	. 8B38	MOV EDI,DWORD PTR DS:[EAX]
00490C12	. FF57 0C	CALL DWORD PTR DS:[EDI+C]
00490C15	. 8B55 E8	MOV EDX,DWORD PTR SS:[EBP-18]
00490C18	. A1 8CAC5200	MOV EAX,DWORD PTR DS:[52AC8C]
00490C1D	. E8 B639F7FF	CALL 3_2.004045D8
00490C22	.v75 0A	JNZ SHORT 3_2.00490C2E
00490C24	. C705 7CAC5200	MOV DWORD PTR DS:[52AC7C],-1
00490C2E	> 46	INC ESI
00490C2F	. 66:FFCB	DEC BX
00490C32	.^75 C7	JNZ SHORT 3_2.00490BF8
00490C34	> 33C0	MOV EAX,0
00490C36	. 5A	POP EDX
00490C37	. 5A	POP ECX

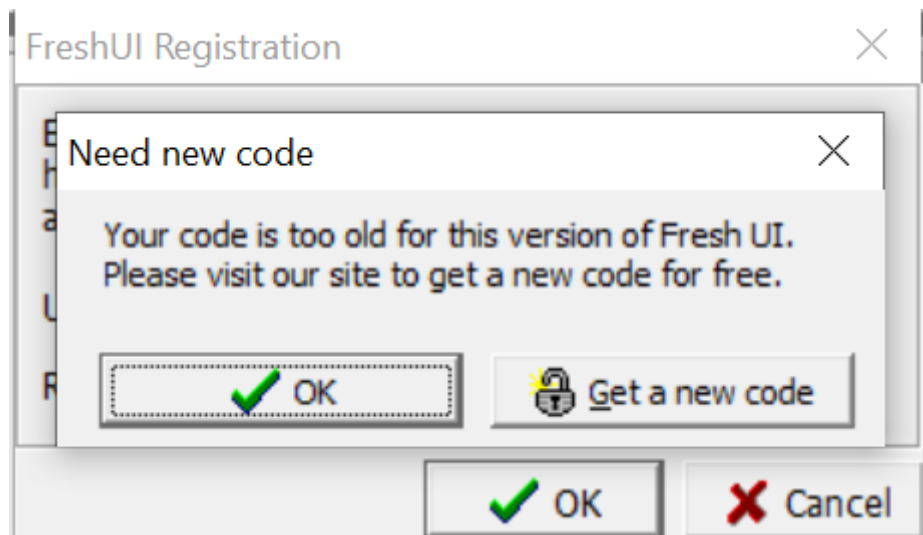
- Đây chính là đoạn code kiểm tra giá trị key nhập vào có khớp với các key có sẵn trong chương trình hay không. Do đó, chúng quyết định việc đoạn code tiếp theo có quay lại GOOD BOY hay không.

00490CF6	. E8 C1FDFFFF	CALL 3_2.00490ABC	
00490CFB	. 833D 80AC5200	CMP DWORD PTR DS:[52AC80],0	
00490D02	. 74 0F	JE SHORT 3_2.00490D13	
00490D04	. A1 48985200	MOV EAX,DWORD PTR DS:[529848]	
00490D09	. 8B00	MOV EAX,DWORD PTR DS:[EAX]	
00490D0B	. 8B10	MOV EDX,DWORD PTR DS:[EAX]	
00490D0D	. FF92 F8000000	CALL DWORD PTR DS:[EDX+F8]	
00490D13	> 833D 78AC5200	CMP DWORD PTR DS:[52AC78],1	
00490D1A	. 1BC0	SBB EAX,EAX	
00490D1C	. 40	INC EAX	
00490D1D	. 3C 01	CMP AL,1	
00490D1F	. 74 12	JE SHORT 3_2.00490D33	
00490D21	. 833D 7CAC5200	CMP DWORD PTR DS:[52AC7C],1	
00490D28	. 1BC0	SBB EAX,EAX	
00490D2A	. 40	INC EAX	
00490D2B	. 3C 01	CMP AL,1	
00490D2D	. 0F85 B4010000	JNZ 3_2.00490EE7	
00490D33	> B8 6C0F4900	MOV EAX,3_2.00490F6C	ASCII "FreshUI has been registered succes
00490D38	. E8 0767FBFF	CALL 3_2.00447444	
00490D3D	. B2 01	MOV DL,1	
00490D3F	. A1 E0464200	MOV EAX,DWORD PTR DS:[4246E0]	
00490D44	. E8 973AF9FF	CALL 3_2.004247E0	
00490D49	. A3 84AC5200	MOV DWORD PTR DS:[52AC84],EAX	

- Như đã trình bày ở trên, ba ô nhớ **[52AC80]**, **[52AC78]** và **[52AC7C]** là những ô nhớ lưu kết quả so sánh key từng loại với registration code người dùng nhập vào. Ở đây sau khi thực hiện hàm tại dòng lệnh **3_2.00490ABC**, ta có 3 lệnh **CMP** tương ứng để kiểm tra xem liệu registration code có khớp với key của hệ thống hay không.
- Sau khi thực hiện lệnh **CALL 3_2.00490ABC**, kết quả sẽ trả về 3 ô nhớ **[52AC80]**, **[52AC78]** và **[52AC7C]**. Tuy nhiên giá trị ô nhớ không phải là cái quyết định registration code ta nhập đúng hay sai, mà là giá trị **AL**.
- Lệnh **CMP <ô nhớ>, 1** chỉ để tác động tới cờ **CF**. Nếu **ô nhớ** có giá trị 1, cờ **CF = 0**. Do đó **SBB EAX, EAX** cho kết quả **EAX = 0**. Tiếp tục thực hiện lệnh **INC EAX** cho kết quả **EAX = 1**. Suy ra **AL = 1**. Lúc đó ta mới tới được lệnh thông báo thành công.
- Chú ý là các ô nhớ không hề được gán giá trị 1 trong bất cứ dòng lệnh nào, nếu registration code ta nhập đúng thì nó sẽ lưu giá trị thành -1. Khi dùng lệnh **CMP <ô nhớ>, 1**, nếu ô nhớ có giá trị -1 thì không thay đổi cờ **CF**, từ đó **AL = 1** và chương trình vẫn sẽ thông báo ta nhập đúng.
- Ta đã biết có 3 loại key trong chương trình nhưng chưa biết cụ thể đó là những loại key gì. Tìm trong bảng Search for all referenced text strings, ta thấy có dòng sau:

Address	Disassembly	Text string
00490E9E	MOV EDX,3_2.00491064	ASCII "About - [Personal License]"
00490EB8	MOV EDX,3_2.00491088	ASCII "FreshUI - [Business License]"
00490ECB	MOV EDX,3_2.004910B0	ASCII "FreshUI - [Personal License]"

- Với những thông tin đã có, ta có thể suy ra: Chương trình có 3 loại key bao gồm **Personal**, **Business** và 1 loại key khác sẽ được đề cập đến sau.
- Quay trở lại đoạn code sau khi gọi lệnh **CALL 3_2.00490ABC** (hàm kiểm tra registration code ta nhập vào với key của hệ thống). Đặt breakpoint ở lệnh **CMP [52AC80], 0** và cho chương trình chạy tới lệnh này, sau đó tiếp tục chạy tới lệnh **JE SHORT 3_2.00490D13**. Thay đổi cờ **ZF** bằng cách nhấn đúp vào giá trị cờ **ZF**. Như vậy lệnh **JE** sẽ không được thực hiện, tức là ta đang giả sử registration code đúng ở loại key này. Tiếp tục nhấn **F7**, chương trình FreshUI hiện ra thông báo như sau:



- Vậy loại key cuối cùng là **Old** và do ô nhớ **[52AC80]** đánh dấu.
- Bây giờ ta sẽ tìm cách load key hệ thống và tìm tập hợp các key hệ thống trong chương trình. Thực hiện lại các bước đặt breakpoint tại lệnh **CALL 3_2.00490ABC**, nhấn F9, nhập User Name, Registration Code và nhấn F7 để chạy vào lệnh CALL. Đặt breakpoint ở tất cả các câu lệnh **CALL DWORD PTR [EDX+14]** để xem số lượng key mỗi loại là bao nhiêu, nhấn F9 để chạy tới dòng đó, sau đó nhấn F8.

00490B35	• 8B80 38020000	MOV EAX, DWORD PTR DS:[EAX+238]
00490B36	• 8B10	MOV EDX, DWORD PTR DS:[EAX]
00490B37	• FFS2 14	CALL DWORD PTR DS:[EDX+14]
00490B40	• 8B08	MOV EBX, EAX
00490B42	• 4B	DEC EBX
00490B43	• 66:850B	TEST BX, BX
00490B46	• 7C 3C	JL SHORT 3_2.00490B84
00490B48	• 43	INC EBX
00490B49	• 33F6	XOR ESI, ESI
00490B4B	• 8D4D F0	LEA ECX, DWORD PTR SS:[EBP-10]
00490B4E	• 0BFDD6	MOVSX EDX, SI
00490B51	• 8B45 FC	MOV EAX, DWORD PTR SS:[EBP-4]
00490B54	• 8B80 64030000	MOV EAX, DWORD PTR DS:[EAX+364]
00490B5A	• 8B80 38020000	MOV EAX, DWORD PTR DS:[EAX+238]
00490B60	• 8B38	MOV EDI, DWORD PTR DS:[EAX]
00490B62	• FFS7 0C	CALL DWORD PTR DS:[EDI+C]
00490B65	• 8B55 F0	MOV EDX, DWORD PTR SS:[EBP-10]
00490B68	• A1 8CAC5200	MOV EAX, DWORD PTR DS:[52AC8C]
00490B6D	• E8 663AF7FF	CALL 3_2.004045D8
00490B72	• 75 0A	JNZ SHORT 3_2.00490B7E
00490B74	• C705 80AC5200	MOV DWORD PTR DS:[52AC80], -1
00490B7E	• 46	INC ESI
00490B7F	• 66:FFCB	DEC BX
00490B82	• 75 C7	JNZ SHORT 3_2.00490B48
00490B84	• 8B45 FC	MOV EAX, DWORD PTR SS:[EBP-4]
00490B87	• 8B80 5C030000	MOV EAX, DWORD PTR DS:[EAX+35C]
00490B8D	• 8B80 38020000	MOV EAX, DWORD PTR DS:[EAX+238]
00490B93	• 8B10	MOV EDX, DWORD PTR DS:[EAX]
00490B95	• FFS2 14	CALL DWORD PTR DS:[EDX+14]
00490B98	• 8B08	MOV EBX, EAX
00490B9A	• 4B	DEC EBX
00490B9B	• 66:850B	TEST BX, BX
00490B9E	• 7C 3C	JL SHORT 3_2.00490BDC
00490BA0	• 43	INC EBX
00490BA1	• 33F6	XOR ESI, ESI
00490BA3	• 8D4D EC	LEA ECX, DWORD PTR SS:[EBP-14]
00490BA6	• 0BFDD6	MOVSX EDX, SI
00490BA9	• 8B45 FC	MOV EAX, DWORD PTR SS:[EBP-4]
00490BAC	• 8B80 5C030000	MOV EAX, DWORD PTR DS:[EAX+35C]
00490BB2	• 8B80 38020000	MOV EAX, DWORD PTR DS:[EAX+238]
00490BB8	• 8B38	MOV EDI, DWORD PTR DS:[EAX]
00490BBA	• FFS7 0C	CALL DWORD PTR DS:[EDI+C]
00490BBD	• 8B55 EC	MOV EDX, DWORD PTR SS:[EBP-14]
00490BC0	• A1 8CAC5200	MOV EAX, DWORD PTR DS:[52AC8C]
00490BC5	• E8 0E3AF7FF	CALL 3_2.004045D8
00490BCA	• 75 0A	JNZ SHORT 3_2.00490BD6
00490BCC	• C705 78AC5200	MOV DWORD PTR DS:[52AC78], -1
00490BD6	• 46	INC ESI
00490BD7	• 66:FFCB	DEC BX
00490BDA	• 75 C7	JNZ SHORT 3_2.00490BA3
00490BDC	• 8B45 FC	MOV EAX, DWORD PTR SS:[EBP-4]
00490BDF	• 8B80 60030000	MOV EAX, DWORD PTR DS:[EAX+360]
00490BE5	• 8B80 38020000	MOV EAX, DWORD PTR DS:[EAX+238]
00490BE8	• 8B10	MOV EDX, DWORD PTR DS:[EAX]
00490BE9	• FFS2 14	CALL DWORD PTR DS:[EDX+14]
00490BF0	• 8B08	MOV EBX, EAX
00490BF2	• 4B	DEC EBX
00490BF3	• 66:850B	TEST BX, BX
00490BF6	• 7C 3C	JL SHORT 3_2.00490C34
00490BF8	• 43	INC EBX
00490BF9	• 33F6	XOR ESI, ESI
00490BFB	• 8D4D E8	LEA ECX, DWORD PTR SS:[EBP-18]
00490BFE	• 0BFDD6	MOVSX EDX, SI
00490C01	• 8B45 FC	MOV EAX, DWORD PTR SS:[EBP-4]
00490C04	• 8B80 60030000	MOV EAX, DWORD PTR DS:[EAX+360]
00490C0A	• 8B80 38020000	MOV EAX, DWORD PTR DS:[EAX+238]
00490C10	• 8B38	MOV EDI, DWORD PTR DS:[EAX]
00490C12	• FFS7 0C	CALL DWORD PTR DS:[EDI+C]
00490C15	• 8B55 E8	MOV EDX, DWORD PTR SS:[EBP-18]
00490C18	• A1 8CAC5200	MOV EAX, DWORD PTR DS:[52AC8C]
00490C1D	• E8 663AF7FF	CALL 3_2.004045D8
00490C22	• 75 0A	JNZ SHORT 3_2.00490C2E
00490C24	• C705 7CAC5200	MOV DWORD PTR DS:[52AC7C], -1
00490C2E	• 46	INC ESI
00490C2F	• 66:FFCB	DEC BX
00490C32	• 75 C7	JNZ SHORT 3_2.00490BF8
00490C34	• 33C0	XOR EAX, EAX

- Quan sát bên thanh ghi Register ta có kết quả sau

- **CALL DWORD PTR [EDX+14]** đầu tiên

Registers (FPU)	
EAX	00000004
ECX	00000000
EDX	00000000
EBX	022F16DC
ESP	0019EE4C
EBP	0019EE7C
ESI	00473BD8 3_2.00473BD8
EDI	0019F048

→ Key loại **Old** có số lượng là 4 key

- **CALL DWORD PTR [EDX+14]** thứ hai

Registers (FPU)	
EAX	000001F0
ECX	00000000
EDX	00000000
EBX	00000000
ESP	0019EE4C
EBP	0019EE7C
ESI	00000004
EDI	004385CC 3_2.004385CC
EIP	00490B98 3_2.00490B98

→ Key loại **Personal** có 1F0 = 496 key

- **CALL DWORD PTR [EDX+14]** thứ ba

Registers (FPU)	
EAX	000001F4
ECX	00000000
EDX	00000000
EBX	00000000
ESP	0019EE4C
EBP	0019EE7C
ESI	000001F0
EDI	004385CC 3_2.004385CC
EIP	00490BF0 3_2.00490BF0

→ Key loại **Business** có **1F4 = 500** key

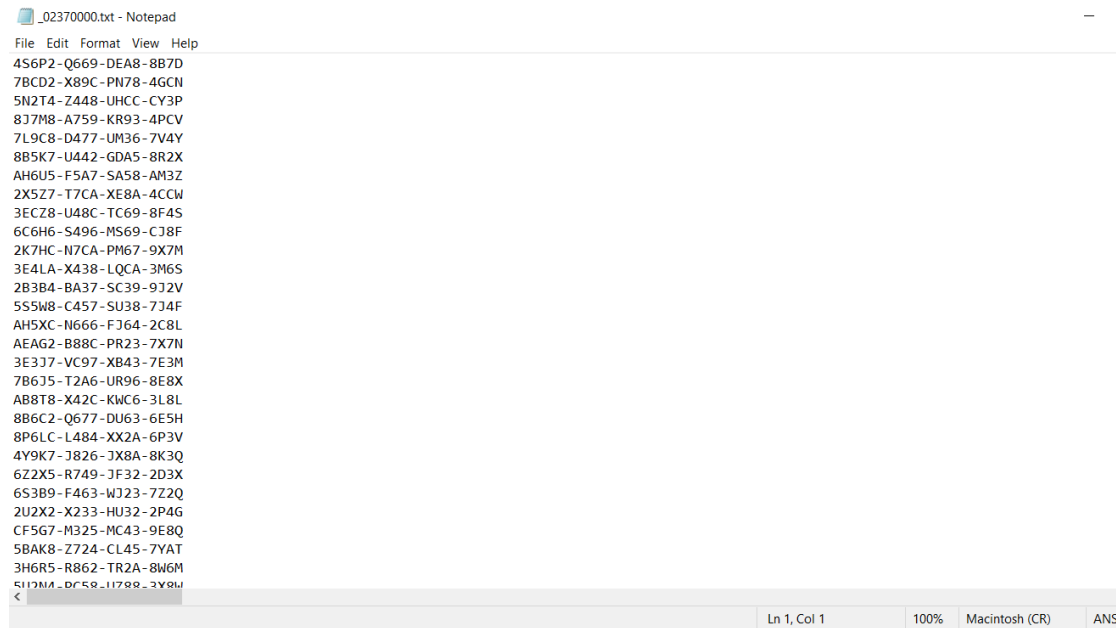
- Để tìm được danh sách các key, ta đặt breakpoint ở dòng lệnh **CALL DWORD PTR [EDI+C]**, nhấn giữ F7 cho đến khi tới được dòng sau

Registers (FPU)			
75E71A89	56	PUSH ESI	
75E71A8A	8B75 08	MOV ESI,DWORD PTR [EBP+8]	
75E71A8D	57	PUSH EDI	
75E71A8E	FFB6 F000	PUSH DWORD PTR [ESI+F0]	
75E71A94	FF36	PUSH DWORD PTR [ESI]	
75E71A96	FF15 540EE	CALL DWORD PTR [75EC0E54]	user32.75E719C0
75E71A9C	8BF8	MOV EDI,EAX	
75E71A9E	8D86 F800	LEA EAX,DWORD PTR [ESI+F8]	
75E71AA4	FF00	INC DWORD PTR [EAX]	
75E71AA6	8338 01	CMP DWORD PTR [EAX],1	

- Nhấn chuột phải vào ô nhớ **EDI**, chọn Follow in Dump, giá trị của ô nhớ đó sẽ hiện ra ở bảng dưới

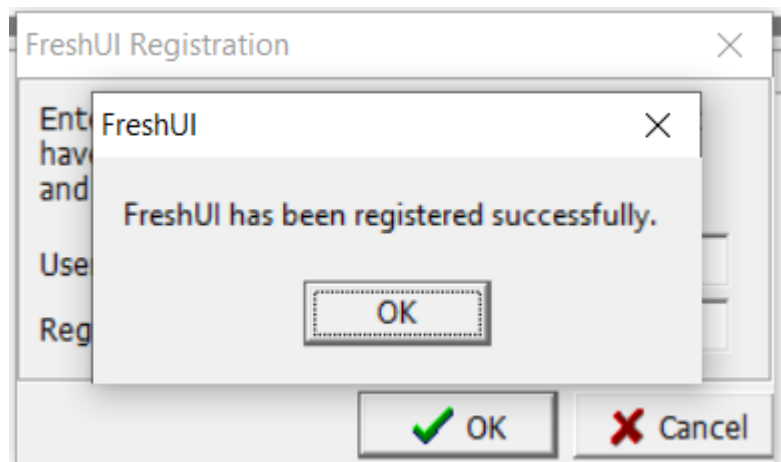
Address	Hex dump												ASCII
002AED78	37	51	35	59	35	2D	56	38	37	38	2D	44	7Q5Y5-V878-DH26-
002AED88	35	52	32	57	0D	0A	38	59	33	5A	38	2D	5R2W..8Y3Z8-C72C
002AED98	2D	4E	46	37	35	2D	35	50	43	52	0D	0A	-NF75-5PCR..AV4K
002AEDA8	34	2D	58	41	38	35	2D	4A	50	41	38	2D	4-XA85-JPA8-AW3E
002AEDB8	0D	0A	37	53	36	44	38	2D	4E	33	35	35	..7S6D8-N355-VYA
002AEDC8	32	2D	35	48	36	47	0D	0A	32	42	33	42	2-5K6G..2B3B9-R5
002AEDD8	35	36	2D	55	58	38	33	2D	41	58	34	55	56-UX83-AX4U..2R
002AEDE8	34	58	34	2D	47	32	35	37	2D	45	58	34	4X4-G257-EX49-8B
002AEDF8	41	58	0D	0A	32	4E	32	5A	32	2D	42	33	AX..2N2Z2-B326-B
002AEE08	44	33	33	2D	34	48	37	4A	0D	0A	34	48	D33-4H7J..4H4U7-
002AEE18	43	32	38	38	2D	47	51	36	41	2D	38	59	C288-GQ6A-8Y4L..
002AEE28	37	52	38	56	35	2D	45	41	36	33	2D	44	7R8V5-EA63-DG2C-
002AEE38	32	58	33	44	0D	0A	32	48	38	58	36	2D	2X3D..2H8X6-MA25
002AEE48	2D	48	4C	38	32	2D	38	42	35	48	0D	0A	-HL82-8B5H..4Y2W
002AEE58	35	2D	42	35	37	33	2D	45	54	35	36	2D	5-B573-ET56-6KAT
002AEE68	0D	0A	32	4D	37	59	35	2D	59	34	34	32	..2M7Y5-Y442-FKA
002AEE78	36	2D	37	58	32	5A	0D	0A	36	5A	37	42	6-7X2Z..6Z7B2-K6
002AEE88	38	33	2D	57	58	35	41	2D	33	50	36	54	83-WX5A-3P6T..3H
002AEE98	41	50	41	2D	45	38	38	37	2D	54	54	33	APA-E887-TT35-7T
002AEEA8	38	44	0D	0A	32	53	41	5A	41	2D	58	32	8D..2SAZA-X299-J
002AEEB8	52	43	41	2D	36	46	34	43	0D	0A	32	4C	RCA-6F4C..2L8S2-
002AEEC8	4E	37	35	38	2D	51	4A	34	35	2D	38	48	N758-QJ45-8H8L..
002AEEED8	33	55	43	57	39	2D	50	33	37	39	2D	58	3UCW9-P379-XF34-
002AEEEE8	39	53	39	53	0D	0A	36	52	37	53	32	2D	9S9S..6R7S2-K336

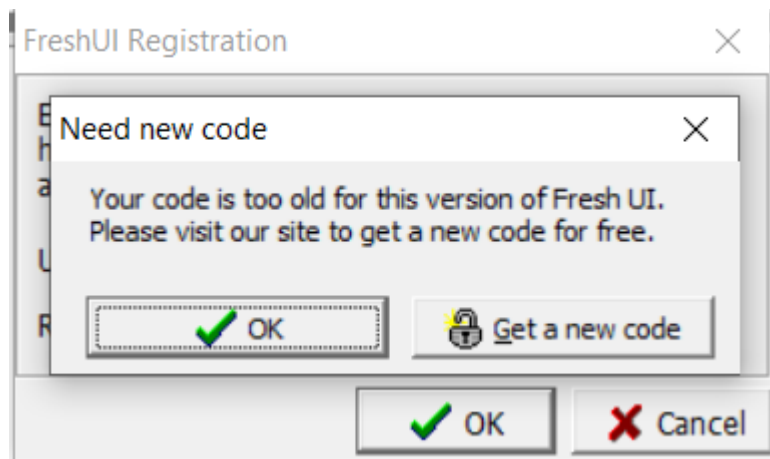
- Đây chính là danh sách các key. Ta xuất ra file bằng cách nhấn chuột phải vào ô nhớ bên cột Address → Backup → Save data to file. OllyDbg sẽ xuất dữ liệu ra cho chúng ta. Vào file vừa xuất ra, ta thấy có danh sách key ở đó



The screenshot shows a Notepad window titled "_02370000.txt - Notepad". The text inside is a list of 48 alphanumeric keys, each on a new line. The keys are: 4S6P2-Q669-DEA8-8B7D, 7BCD2-X89C-PN78-4G6N, 5N2T4-Z448-UHCC-CY3P, 8J7M8-A759-KR93-4PCV, 7L9C8-D477-UM36-7V4Y, 8B5K7-U442-GDA5-8R2X, AH6U5-F5A7-SA58-AM3Z, 2X5Z7-T7CA-XE8A-4CCW, 3ECZ8-U48C-TC69-8F4S, 6C6H6-S496-MS69-C38F, 2K7HC-N7CA-PM67-9X7M, 3E4LA-X438-LQCA-3M6S, 2B3B4-BA37-SC39-9J2V, 5S5W8-C457-SU38-7J4F, AH5XC-N666-FJ64-2C8L, AEAG2-B88C-PR23-7X7N, 3E3J7-VC97-XB43-7E3M, 7B6J5-T2A6-UR96-8E8X, AB8T8-X42C-KWC6-3L8L, 8B6C2-Q677-DU63-6E5H, 8P6LC-L484-XX2A-6P3V, 4Y9K7-J826-JX8A-8K3Q, 6Z2X5-R749-JF32-2D3X, 6S3B9-F463-WJ23-7Z2Q, 2U2X2-X233-HU32-2P4G, CF5G7-M325-MC43-9E8Q, 5BAK8-Z724-CL45-7YAT, 3H6R5-R862-TR2A-8W6M, 5U7MA-DC58-11788-3Y8W. The status bar at the bottom indicates "Ln 1, Col 1", "100%", "Macintosh (CR)", and "ANSI".

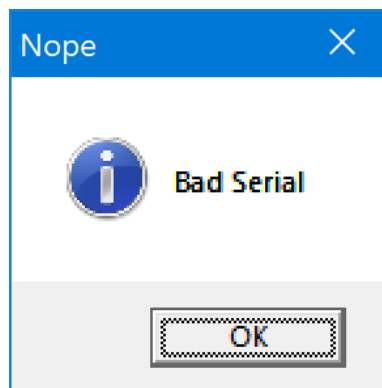
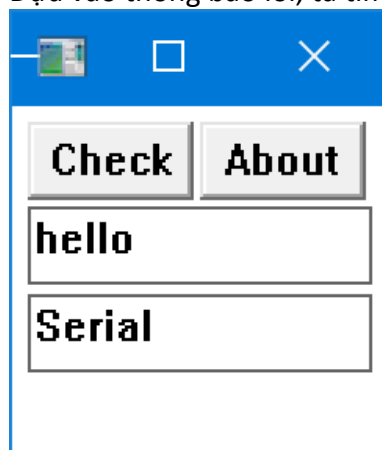
- Nhập 1 trong 3 loại key vào chương trình, ta nhận được các thông báo như sau:





3. Bài 3.3

- Dựa vào thông báo lỗi, ta tìm đến chuỗi **Bad Serial**

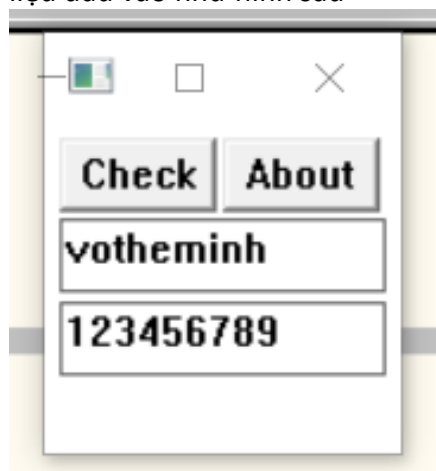


00402475	PUSH	3_3.0041110B	ASCII	"%i"
004024A0	PUSH	3_3.0041110E	ASCII	"Good"
004024B2	PUSH	3_3.00411113	ASCII	"Good, Now Make a Key"
004024C2	PUSH	3_3.0041112C	ASCII	"Nope"
004024C7	PUSH	3_3.00411131	ASCII	"Bad Serial"
004024E4	PUSH	3_3.00411104	ASCII	"OK"

- Đi đến đoạn code liên quan, ta được

0040247E	. 56	PUSH ESI	
00402483	. E8 B5490000	CALL 3_3.00406E38	
00402486	. 8D4D C0	LEA ECX,DWORD PTR SS:[EBP-40]	
00402487	. 51	PUSH ECX	
00402489	. 6A 0E	PUSH 0E	
0040248B	. 6A 0D	PUSH 0D	
00402491	. FF35 80524100	PUSH DWORD PTR DS:[415280]	
00402497	. FF15 40164000	CALL DWORD PTR DS:[<&USER32.SendMessageA	[lParam wParam = E Message = WM_GETTEXT hWnd = 1708D6 SendMessageA
0040249D	. 83C4 10	ADD ESP,10	
0040249E	. 8D45 C0	LEA EAX,DWORD PTR SS:[EBP-40]	
0040249F	. 50	PUSH EAX	
004024A4	. 56	PUSH ESI	
004024A7	. E8 C4470000	CALL 3_3.00406C68	
004024A9	. 83C4 08	ADD ESP,8	
004024AB	. 85C0	TEST EAX,EAX	
004024AD	. 75 15	JNZ SHORT 3_3.004024C0	
004024AE	. 6A 40	PUSH 40	
004024B2	. 68 0E114100	PUSH 3_3.0041110E	
004024B7	. 68 13114100	PUSH 3_3.00411113	
004024B9	. 6A 00	PUSH 0	
004024BE	. E8 B2DC0000	CALL <JMP.&USER32.MessageBoxA>	[Style = MB_OK!MB_ICONASTERISK!MB_AP Title = "Good" Text = "Good, Now Make a Keygen!" hOwner = NULL MessageBoxA
004024C0	. EB 5A	JMP SHORT 3_3.0040251A	
004024C2	. 6A 40	PUSH 40	
004024C7	. 68 2C114100	PUSH 3_3.0041112C	
004024CC	. 68 31114100	PUSH 3_3.00411131	
004024CE	. 6A 00	PUSH 0	
004024D3	. E8 9DDC0000	CALL <JMP.&USER32.MessageBoxA>	[Style = MB_OK!MB_ICONASTERISK!MB_AP Title = "Nope" Text = "Bad Serial" hOwner = NULL MessageBoxA
	. FF 45	JMP SHORT 3_3.0040251A	

- Đặt breakpoint ở dòng **0040247E**, chạy debug từng dòng lệnh. Nhập liệu đầu vào như hình sau



- Sau khi chạy qua lệnh **CALL 3_3.00406E38**, ta thấy giá trị khả nghi trong thanh ghi **EDX** **"803218662"**, **ESI** = **"803218662"**

0040247E	. 8D75 D8	LEA ESI,DWORD PTR SS:[EBP-28]	
0040247D	. 56	PUSH ESI	
0040247E	. E8 B5490000	CALL 3_3.00406E38	
00402483	. 8D4D C0	LEA ECX,DWORD PTR SS:[EBP-40]	
00402486	. 51	PUSH ECX	
00402487	. 6A 0E	PUSH 0E	
00402489	. 6A 0D	PUSH 0D	
0040248B	. FF35 80524100	PUSH DWORD PTR DS:[415280]	
00402491	. FF15 40164000	CALL DWORD PTR DS:[<&USER32.SendMessageA	

Registers (FPU)	
EAX	00000009
ECX	0019FA44
EDX	0019FA5C ASCII "803218662"
EBX	00000006
ESP	0019FA04
EBP	0019FA84
ESI	0019FA5C ASCII "803218662"
EDI	0019FA42
EIP	00402491 3_3.00402491
C 0	ES 002B 32bit 0(FFFFFFFF)
P 1	CS 0023 32bit 0(FFFFFFFF)

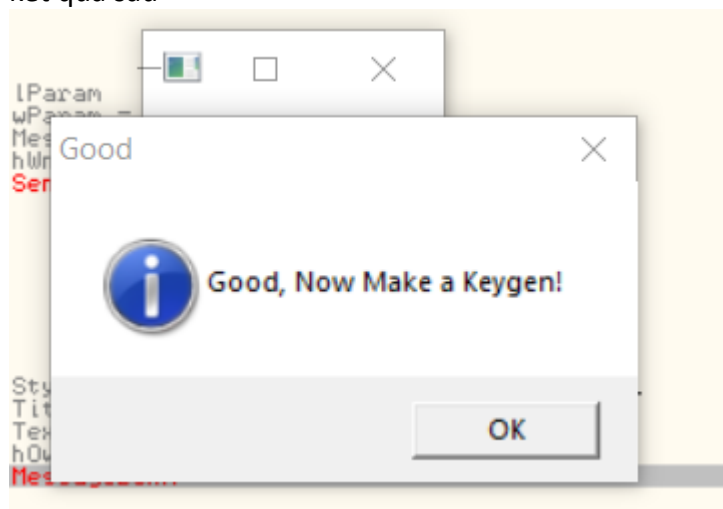
- Tới lệnh **JNZ 3_3.004024C0** là lệnh nhảy đến GOODBOY nên ta có thể đoán là lệnh **TEST EAX, EAX**, EAX là kiểm tra điều kiện chuỗi ta nhập vào

004024A4	. 83C4 08	HUU ESP, 8
004024A7	. 85C0	TEST EAX, EAX
004024A9	. 75 15	JNZ SHORT 3_3.004024C0
004024AB	. 6A 40	PUSH 40
004024AD	. 68 0E114100	PUSH 3_3.0041110E
004024B2	. 68 13114100	PUSH 3_3.00411113
004024B7	. 6A 00	PUSH 0

- **CALL 3_3.004024C0** là lệnh so sánh giá trị trong **EDX = "123456789"** với giá trị trong thành ghi **ECX = "803218662"** nếu đúng thì **EAX = 0**, nếu sai thì **EAX = 1** như hình sau

Registers (FPU)	
EAX	00000001
ECX	0019FA5C ASCII "803218662"
EDX	0019FA44 ASCII "123456789"
EBX	00000006
ESP	0019FA24
EBP	0019FA84
ESI	0019FA5C ASCII "803218662"
EDI	0019FA42
EIP	004024A9 3_3.004024A9
C 0	ES 002B 32bit 0(FFFFFFFF)

- Chạy tiếp tục tiếp theo chắc chắn sẽ chạy vào BAD BOY.
- Nhập lại User Name là **votheminh** và Serial là **803218662**, ta nhận được kết quả sau



- Vậy với Username là **votheminh** thì Serial tương ứng là **803218662**
- Để tạo KeyGen, chúng ta chú ý 3 đoạn code sau

- Đoạn thứ nhất

00402353	. 6A 0D	PUSH 0D	Message = WM_GETTEXT
00402355	. FF35 7C524100	PUSH DWORD PTR DS:[41527C]	hWnd = 807EA
0040235B	. FF15 40164000	CALL DWORD PTR DS:[40164000]	SendMessageA
00402361	. 8D4D B8	LEA ECX, DWORD PTR SS:[EBP-48]	
00402364	. 51	PUSH ECX	

- Lệnh **0040235B** là hàm mã hóa lần 1
- Mã hóa chuỗi ta nhập vào từ ký tự thành **hệ Thập Lục Phân** trong bảng mã ASCII

- Đoạn code thứ 2

0040239C	. C745 F8 000000	MOV DWORD PTR SS:[EBP-8],0
004023A3	> 8D55 B8	LEA EDX,DWORD PTR SS:[EBP-48]
004023A6	. 52	PUSH EDX
004023A7	. E8 70490000	CALL 3_3.00406D1C
004023AC	. 83C4 04	ADD ESP,4
004023AF	. 3B45 F8	CMP EAX,DWORD PTR SS:[EBP-8]
004023B2	✓ 0F8C B7000000	JL 3_3.0040246F
004023B8	. 8B5D F8	MOV EBX,DWORD PTR SS:[EBP-8]
004023BB	. 83FB 08	CMP EBX,8
004023BE	. BE 01000000	MOV ESI,1
004023C3	✓ 72 05	JB SHORT 3_3.004023CA
004023C5	. BE 00000000	MOV ESI,0
004023CA	✓ 72 0A	JB SHORT 3_3.004023D6
004023CC	. B8 A1000000	MOV EAX,0A1
004023D1	. E8 72010000	CALL 3_3.00402548
004023D6	> 8D7C1D B8	LEA EDI,DWORD PTR SS:[EBP+EBX-48]
004023DA	. 803F 00	CMP BYTE PTR DS:[EDI],0
004023DD	✓ 0F84 84000000	JE 3_3.00402467
004023E3	. 85F6	TEST ESI,ESI
004023E5	✓ 75 0A	JNZ SHORT 3_3.004023F1
004023E7	. B8 A1000000	MOV EAX,0A1
004023EC	. E8 57010000	CALL 3_3.00402548
004023F1	> 803F 00	CMP BYTE PTR DS:[EDI],0
004023F4	✓ 0F84 6D000000	JE 3_3.00402467
004023FA	. 8B4D F8	MOV ECX,DWORD PTR SS:[EBP-8]
004023FD	. 83F9 08	CMP ECX,8
00402400	. BA 01000000	MOV EDX,1
00402405	✓ 72 02	JB SHORT 3_3.00402409
00402407	. 8AD6	MOV DL,DH
00402409	> 8955 B4	MOV DWORD PTR SS:[EBP-4C],EDX
0040240C	✓ 72 0A	JB SHORT 3_3.00402418
0040240E	. B8 A2000000	MOV EAX,0A2
00402413	. E8 30010000	CALL 3_3.00402548
00402418	> 8B5D F8	MOV EBX,DWORD PTR SS:[EBP-8]
0040241B	. 8D741D B8	LEA ESI,DWORD PTR SS:[EBP+EBX-48]
0040241F	. 0FB606	MOVZX EAX,BYTE PTR DS:[ESI]
00402422	. B9 05000000	MOV ECX,5
00402427	. 99	CDQ
00402428	. F7F9	IDIV ECX
0040242A	. 89C3	MOV EBX,EAX
0040242C	. 89D9	MOV ECX,EBX
0040242E	. C1F9 1F	SAR ECX,1F
00402431	. 8B45 F0	MOV EAX,DWORD PTR SS:[EBP-10]
00402434	. 8B55 F4	MOV EDX,DWORD PTR SS:[EBP-C]
00402437	. 0FAFC8	IMUL ECX,EAX
0040243A	. 0FAFD3	IMUL EDX,EBX
0040243D	. 03CA	ADD ECX,EDX
0040243F	. F7E3	MUL EBX
00402441	. 03D1	ADD EDX,ECX
00402443	. 8945 F0	MOV DWORD PTR SS:[EBP-10],EAX
00402446	. 8955 F4	MOV DWORD PTR SS:[EBP-C],EDX
00402449	. 837D B4 00	CMP DWORD PTR SS:[EBP-4C],0
0040244D	✓ 75 0A	JNZ SHORT 3_3.00402459
0040244F	. B8 A3000000	MOV EAX,0A3
00402454	. E8 EF000000	CALL 3_3.00402548
00402459	> 0FB63E	MOVZX EDI,BYTE PTR DS:[ESI]
0040245C	. 89FA	MOV EDX,EDI
0040245E	. C1FA 1F	SAR EDX,1F
00402461	. 017D F0	ADD DWORD PTR SS:[EBP-10],EDI
00402464	. 1155 F4	ADC DWORD PTR SS:[EBP-C],EDX
00402467	> FF45 F8	INC DWORD PTR SS:[EBP-8]
0040246A	✓ E9 34FFFFFF	JMP 3_3.004023A3
0040246F	> FF75 F4	PUSH DWORD PTR SS:[EBP-C]

- Các lệnh từ dòng **004023A3** đến **0040246A** là tập các hàm mã hóa lần 2
- Mã hóa **chuỗi Thập Lục Phân** từ hàm mã hóa lần 1 thành 1 **chuỗi thập lục phân mới**, thuật toán đã được trình bày trong phần phụ lục.

• Đoạn code thứ 3

0040248B	. FF35 80524100	PUSH DWORD PTR DS:[415280]
00402491	. FF15 40164000	CALL DWORD PTR DS:[40164000]
00402497	. 83C4 10	ADD ESP,10

- Dòng lệnh **00402491** gọi đến hàm mã hóa lần 3
- Mã hóa từ chuỗi thập lục phân từ hàm mã hóa lần 2 thành 1 chuỗi thập lục phân mới, thuật toán đã được trình bày trong phần phụ lục.

- Hình ảnh đoạn code mã hoá lần 3

0040774F	. 8945 F0	MOV DWORD PTR SS:[EBP-10],EAX
00407752	> 83BD 6CFFFFFF	CMP DWORD PTR SS:[EBP-94],0
00407759	. 74 1F	JE SHORT 3_3.0040777A
0040775B	. 8B55 F0	MOV EDX,DWORD PTR SS:[EBP-10]
0040775E	. 8D4D C7	LEA ECX,DWORD PTR SS:[EBP-39]
00407761	. 3BD1	CMP EDX,ECX
00407763	. 75 15	JNZ SHORT 3_3.0040777A
00407765	. 8B45 90	MOV EAX,DWORD PTR SS:[EBP-70]
00407768	. 31D2	XOR EDX,EDX
0040776A	. BB 3A000000	MOV EBX,3A
0040776F	. 8945 88	MOV DWORD PTR SS:[EBP-78],EAX
00407772	. 8955 8C	MOV DWORD PTR SS:[EBP-74],EDX
00407775	. 8955 90	MOV DWORD PTR SS:[EBP-70],EDX
00407778	. EB 65	JMP SHORT 3_3.004077DF
0040777A	> 56	PUSH ESI
0040777B	. 8B5D EC	MOV EBX,DWORD PTR SS:[EBP-14]
0040777E	. 89D9	MOV ECX,EBX
00407780	. 8B55 8C	MOV EDX,DWORD PTR SS:[EBP-74]
00407783	. 8B45 88	MOV EAX,DWORD PTR SS:[EBP-78]
00407786	. C1F9 1F	SAR ECX,1F
00407789	. 57	PUSH EDI
0040778A	. E8 890A0000	CALL 3_3.00408218
0040778F	. 83C3 30	ADD EBX,30
00407792	. 5F	POP EDI
00407793	. 5E	POP ESI
00407794	. 83D1 00	ADC ECX,0
00407797	. 8B5D F4	MOV BYTE PTR SS:[EBP-C],BL
0040779A	. 80FB 39	CMP BL,39
0040779D	. 7E 20	JLE SHORT 3_3.004077BF
0040779F	. F7C6 00010000	TEST ESI,100
004077A5	. 8B45 F0	MOV EAX,DWORD PTR SS:[EBP-10]
004077A8	. 8A5D F4	MOV BL,BYTE PTR SS:[EBP-C]
004077AB	. BA 07000000	MOV EDX,7
004077B0	. 75 05	JNZ SHORT 3_3.004077B7
004077B2	. BA 27000000	MOV EDX,27
004077B7	> 00D3	ADD BL,DL
004077B9	. 8945 F0	MOV DWORD PTR SS:[EBP-10],EAX
004077BC	. 8B5D F4	MOV BYTE PTR SS:[EBP-C],BL
004077BF	> 56	PUSH ESI
004077C0	. 8B5D EC	MOV EBX,DWORD PTR SS:[EBP-14]
004077C3	. 89D9	MOV ECX,EBX
004077C5	. 8B55 8C	MOV EDX,DWORD PTR SS:[EBP-74]
004077C8	. 8B45 88	MOV EAX,DWORD PTR SS:[EBP-78]
004077CB	. C1F9 1F	SAR ECX,1F
004077CE	. 57	PUSH EDI
004077CF	. E8 440A0000	CALL 3_3.00408218
004077D4	. 5F	POP EDI
004077D5	. 5E	POP ESI
004077D6	. 8A5D F4	MOV BL,BYTE PTR SS:[EBP-C]
004077D9	. 8945 88	MOV DWORD PTR SS:[EBP-78],EAX
004077DC	. 8955 8C	MOV DWORD PTR SS:[EBP-74],EDX
004077DF	> 8B4D F0	MOV ECX,DWORD PTR SS:[EBP-10]
004077E2	. 8B45 90	MOV EAX,DWORD PTR SS:[EBP-70]
004077E5	. 31D2	XOR EDX,EDX
004077E7	. FF4D F0	DEC DWORD PTR SS:[EBP-10]
004077EA	. 0B45 88	OR EAX,DWORD PTR SS:[EBP-78]
004077ED	. 0B55 8C	OR EDX,DWORD PTR SS:[EBP-74]
004077F0	. 09C2	OR EDX,EAX
004077F2	. 8B19	MOV BYTE PTR DS:[ECX],BL
004077F4	. 0F85 58FFFFFF	JNZ 3_3.00407752
004077FA	. EB 0E	JMP SHORT 3_3.0040780A

IV. Phụ lục: Cách tạo Serial từ Username bài 3_3

1. Bước 1

- Username đc chuẩn hóa từng byte về dạng mã ASCII hệ 16 (ví dụ Namee = 4E616D6565)

2. Bước 2: Vòng lặp mã hoá lần 2

- 19FA74 = lưu kết quả
- EAX = Lấy từng byte của username(65, 65, 6D, 61, 4E)
- EBX = EAX / 5
- EAX = 19FA74
- EAX = EAX * EBX

- EDI = Lấy từng byte của username(65, 65, 6D, 61, 4E)
 - $19FA74 = EAX + EDI$
 - Kết thúc vòng lặp khi lặp đúng với số kí tự của username (tối thiểu là 5, tối đa là 6), kết quả của vòng lặp này chứa trong **19FA74**
 - Kiểm tra dấu của **[19FA74]**, nếu mang giá trị âm thì lấy bù 2, nếu là dương thì giữ nguyên
3. Bước 3: Vòng lặp lần 3, mã hoá tạo ra Serial
- $[19FA74] \% A = x$
 - $[19FA74] = [19FA74] / A$
 - Kết thúc vòng lặp khi **[19FA74] = 0**, giá trị của x được lưu như key tương ứng với username người dùng nhập vào.
4. Ví dụ: Người dùng nhập Username: “Namee”
- Lần mã hoá đầu tiên: N = 4E, a = 61, m = 6D, e = 65, e = 65
 - Kết quả trả về là **4E616D6565**
 - Lần mã hoá thứ 2
 - **Vòng lặp 1**
 - $19FA74 = 23E$
 - $EAX = 4E$
 - $EBX = EAX / 5 = F$
 - $EAX = 23E$
 - $EAX = EAX * EBX = 23E * F = 21A2$
 - $19FA74 = EAX = 21A2$
 - $EDI = 4E$
 - $19FA74 = 19FA74 + EDI = 21A2 + 4E = 21F0$
 - **Vòng lặp 2**
 - $19FA74 = 21F0$
 - $EAX = 61$
 - $EBX = EAX / 5 = 61 / 5 = 13$
 - $EAX = 21F0$
 - $EAX = EAX * EBX = 21F0 * 13 = 284D0$
 - $19FA74 = EAX = 284D0$
 - $EDI = 61$
 - $19FA74 = 284D0 + 61 = 28531$
 - **Vòng lặp 3**
 - $19FA74 = 28531$
 - $EAX = 6D$
 - $EBX = EAX / 5 = 6D / 5 = 15$
 - $EAX = 28531$
 - $EAX = 28531 * 15 = 34ED05$
 - $19FA74 = EAX = 34ED05$
 - $EDI = 6D$
 - $19FA74 = 19FA74 + 6D = 34ED72$
 - **Vòng lặp 4**
 - $19FA74 = 34ED72$
 - $EAX = 65$
 - $EBX = EAX / 5 = 65 / 5 = 14$
 - $EAX = 19FA74 = 34ED72$

- $EAX = EAX * EBX = 34ED72 * 14 = 4228CE8$
- $19FA74 = EAX 4228CE8$
- $EDI = 65$
- $19FA74 = 19FA74 + 65 = 4228D4D$
- **Vòng lặp 5**
 - $19FA74 = 4228D4D$
 - $EAX = 65$
 - $EBX = EAX / 5 = 65 / 5 = 14$
 - $EAX = 19FA74 = 4228D4D$
 - $EAX = EAX * EBX = 4228D4D * 14 = 52B30A04$
 - $19FA74 = 52B30A04$
 - $EDI = 65$
 - $19FA74 = 19FA74 + 65 = 52B30A69$
- Kết quả trả về là **52B30A69 > 0** nên không cần lấy bù 2
- Lần mã hoá thứ 3
 - $52B30A69 / A = 8451AA4, 52B30A69 \% A = 1$
 - $8451AA4 / A = D3B5DD, 8451AA4 \% A = 2$
 - $D3B5DD / A = 152BC9, D3B5DD \% A = 3$
 - $152BC9 / A = 21DFA, 152BC9 \% A = 5$
 - $21DFA / A = 3632, 21DFA \% A = 6$
 - $3632 / A = 56B, 3632 \% A = 4$
 - $56B / A = 8A, 56B \% A = 7$
 - $8A / A = D, 8A \% A = 8$
 - $D / A = 1, D \% A = 3$
 - $1 / A = 0, 1 \% A = 1$
 - Serial thu được bằng cách viết ngược lại các kết quả sinh ra trong quá trình tính toán tại lần mã hoá thứ 3.
 - **Tóm lại: Đối với Username = "Namee" thì Serial tương ứng là 1387465321**