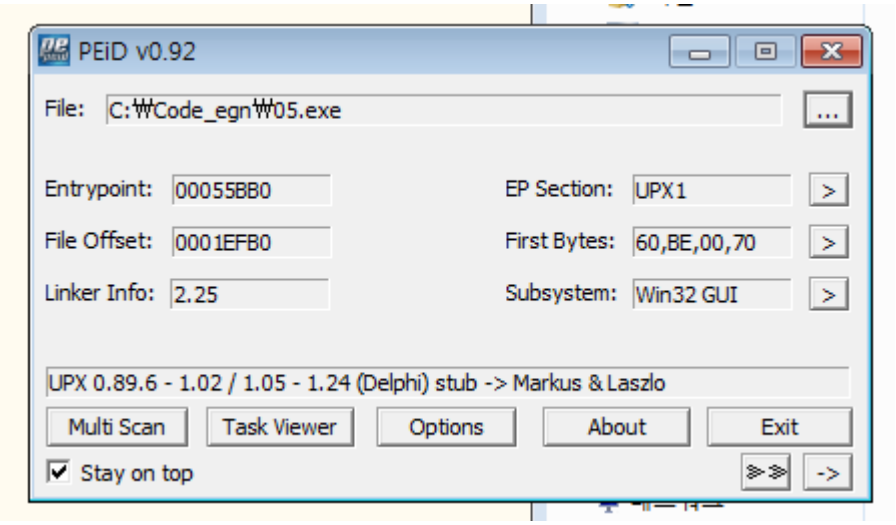
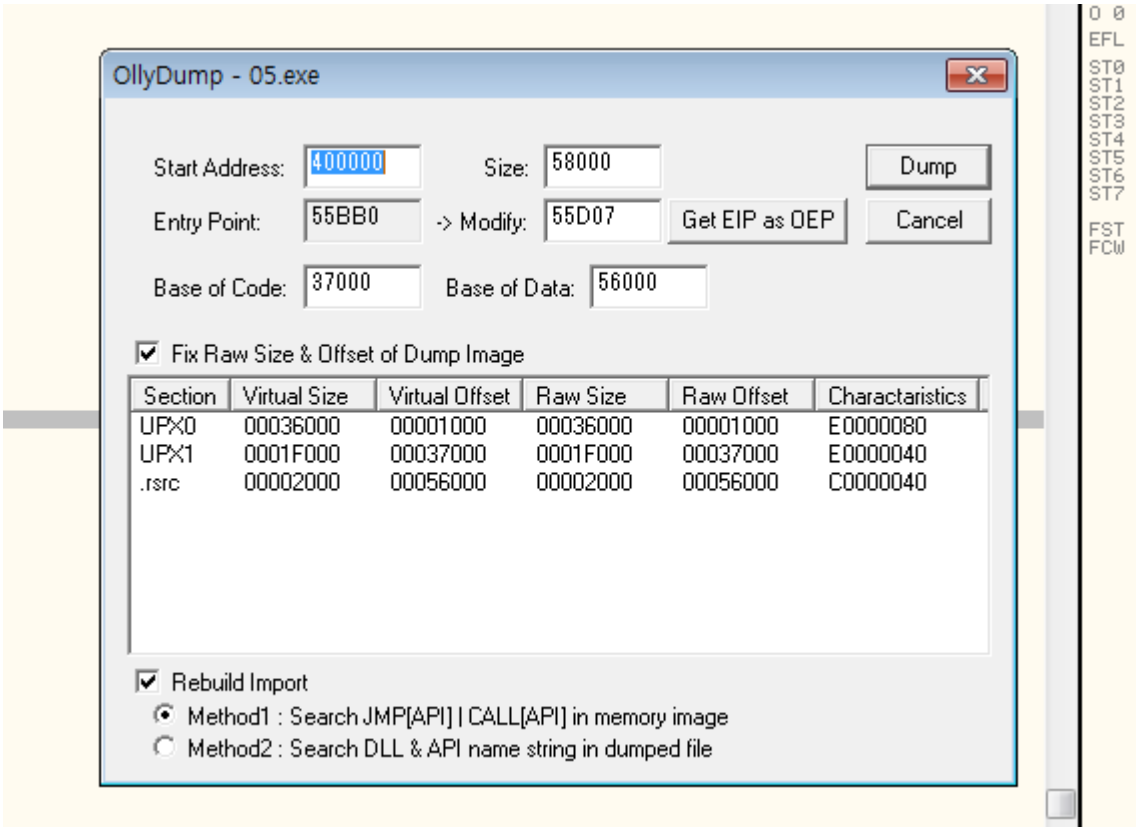
Code Egn 5번문제 풀이



UPX 패킹이 되어 있네요



Olly로 적당히 풀었습니다

```
00440EA0 DD 05_UPXUn.00433584                ASCII "?C"
00440EA7 ASCII "Unit1"
00440EDC MOV ECX,05_UPXUn.00440FC8           ASCII "No Name entered"
00440EE1 MOV EDX,05_UPXUn.00440FD8           ASCII "Enter a Name!"
00440F08 MOV ECX,05_UPXUn.00440FE8           ASCII "No Serial entered"
00440F0D MOV EDX,05_UPXUn.00440FFC           ASCII "Enter a Serial!"
00440F2F MOV EDX,05_UPXUn.00441014           ASCII "Registered User"
00440F4C MOV EDX,05_UPXUn.0044102C           ASCII "GFX-754-IER-954"
00440F5A MOV ECX,05_UPXUn.0044103C           ASCII "CrackMe cracked successfully"
00440F5F MOV EDX,05_UPXUn.0044105C           ASCII "Congrats! You cracked this CrackMe!"
00440F74 MOV ECX,05_UPXUn.00441080           ASCII "Beggar off!"
00440F79 MOV EDX,05_UPXUn.0044108C           ASCII "Wrong Serial,try again!"
00440F8E MOV ECX,05_UPXUn.00441080           ASCII "Beggar off!"
00440F93 MOV EDX,05_UPXUn.0044108C           ASCII "Wrong Serial,try again!"
00440FC8 ASCII "No Name entered",0
00440FD8 ASCII "Enter a Name!",0
00440FE8 ASCII "No Serial entere"
00440FF8 ASCII "d",0
00440FFC ASCII "Enter a Serial!",0
00441014 ASCII "Registered User",0
0044102C ASCII "GFX-754-IER-954",0
```

문자열 중에 축하합니다! 가 많이 의심스러워 거기로 날아가 보았습니다

```
00440EEB . 8B00          MOV EAX,DWORD PTR DS:[EAX]
00440EED . E8 76C1FFFF   CALL 05_UPXUn.0043D068
00440EF2 > 8D55 FC       LEA EDX,DWORD PTR SS:[EBP-4]
00440EF5 . 8B83 C802000( MOV EAX,DWORD PTR DS:[EBX+2C8]
00440EFB . E8 20FFFDFF   CALL 05_UPXUn.00420E20
00440F00 . 837D FC 00    CMP DWORD PTR SS:[EBP-4],0
00440F04 .v 75 18        JNZ SHORT 05_UPXUn.00440F1E
00440F06 . 6A 00         PUSH 0
00440F08 . B9 E80F4400   MOV ECX,05_UPXUn.00440FE8        ASCII "No Serial entered"
00440F0D . BA FC0F4400   MOV EDX,05_UPXUn.00440FFC        ASCII "Enter a Serial!"
00440F12 . A1 442C4400   MOV EAX,DWORD PTR DS:[442C44]
00440F17 . 8B00          MOV EAX,DWORD PTR DS:[EAX]
00440F19 . E8 4AC1FFFF   CALL 05_UPXUn.0043D068
00440F1E > 8D55 FC       LEA EDX,DWORD PTR SS:[EBP-4]
00440F21 . 8B83 C402000( MOV EAX,DWORD PTR DS:[EBX+2C4]
00440F27 . E8 F4FEFDFF   CALL 05_UPXUn.00420E20
00440F2C . 8B45 FC       MOV EAX,DWORD PTR SS:[EBP-4]
00440F2F . BA 14104400   MOV EDX,05_UPXUn.00441014        ASCII "Registered User"
00440F34 . E8 F32BFCFF   CALL 05_UPXUn.00403B2C
00440F39 .v 75 51        JNZ SHORT 05_UPXUn.00440F8C
00440F3B . 8D55 FC       LEA EDX,DWORD PTR SS:[EBP-4]
00440F3E . 8B83 C802000( MOV EAX,DWORD PTR DS:[EBX+2C8]
00440F44 . E8 D7FEFDFF   CALL 05_UPXUn.00420E20
00440F49 . 8B45 FC       MOV EAX,DWORD PTR SS:[EBP-4]
00440F4C . BA 2C104400   MOV EDX,05_UPXUn.0044102C        ASCII "GFX-754-IER-954"
00440F51 . E8 D62BFCFF   CALL 05_UPXUn.00403B2C
00440F56 .v 75 1A        JNZ SHORT 05_UPXUn.00440F72
00440F58 . 6A 00         PUSH 0
00440F5A . B9 3C104400   MOV ECX,05_UPXUn.0044103C        ASCII "CrackMe cracked successfully"
00440F5F . BA 5C104400   MOV EDX,05_UPXUn.0044105C        ASCII "Congrats! You cracked this CrackMe!"
00440F64 . A1 442C4400   MOV EAX,DWORD PTR DS:[442C44]
00440F69 . 8B00          MOV EAX,DWORD PTR DS:[EAX]
00440F6B . E8 F8C0FFFF   CALL 05_UPXUn.0043D068
00440F70 .v EB 32        JMP SHORT 05_UPXUn.00440FA4
00440F72 > 6A 00         PUSH 0
00440F74 . B9 80104400   MOV ECX,05_UPXUn.00441080        ASCII "Beggar off!"
00440F79 . BA 8C104400   MOV EDX,05_UPXUn.0044108C        ASCII "Wrong Serial,try again!"
00440F7E . A1 442C4400   MOV EAX,DWORD PTR DS:[442C44]
00440F83 . 8B00          MOV EAX,DWORD PTR DS:[EAX]
00440F85 . E8 DEC0FFFF   CALL 05_UPXUn.0043D068
00440F8A .v EB 18        JMP SHORT 05_UPXUn.00440FA4
00440F8C > 6A 00         PUSH 0
00440F8E . B9 80104400   MOV ECX,05_UPXUn.00441080        ASCII "Beggar off!"
00440F93 . BA 8C104400   MOV EDX,05_UPXUn.0044108C        ASCII "Wrong Serial,try again!"
00440F98 . A1 442C4400   MOV EAX,DWORD PTR DS:[442C44]
00440F9D . 8B00          MOV EAX,DWORD PTR DS:[EAX]
```

의심스러운 문자열들이 마~ㄴ이 있습니다.

그중 이름부분인듯 보이는 registered user밑의 함수로 들어가 봅니다

```
00403B2F   89C6          MOV ESI,EAX
00403B31   89D7          MOV EDI,EDX
00403B33   39D0          CMP EAX,EDX
00403B35 v 0F84 8F000000 JE 05_UPXUn.00403BCA
00403B3B   85F6          TEST ESI,ESI
00403B3D v 74 68         JE SHORT 05_UPXUn.00403BA7
00403B3F   85FF          TEST EDI,EDI
00403B41 v 74 6B         JE SHORT 05_UPXUn.00403BAE
00403B43   8B46 FC       MOV EAX,DWORD PTR DS:[ESI-4]
00403B46   8B57 FC       MOV EDX,DWORD PTR DS:[EDI-4]
00403B49   29D0          SUB EAX,EDX
```

받아들인 입력값과 registered user를 각각 ESI, EDI에 넣은뒤에 TEST로 문자의 공백여부를 확인

이후에 각 문자열의 길이를 확인 후 EAX – EDX의 절대값 연산을 합니다.

```
00403B4F   52            PUSH EDX
00403B50   C1EA 02       SHR EDX,2
00403B53 v 74 26         JE SHORT 05_UPXUn.00403B7B
00403B55   8B0E          MOV ECX,DWORD PTR DS:[ESI]
00403B57   8B1F          MOV EBX,DWORD PTR DS:[EDI]
00403B59   39D9          CMP ECX,EBX
00403B5B v 75 58         JNZ SHORT 05_UPXUn.00403BB5
00403B5D   4A            DEC EDX
00403B5E v 74 15         JE SHORT 05_UPXUn.00403B75
00403B60   8B4E 04       MOV ECX,DWORD PTR DS:[ESI+4]
00403B63   8B5F 04       MOV EBX,DWORD PTR DS:[EDI+4]
```

그후에 EDX에 EAX값을 더한 뒤 SHR연산을 2회 실행하고 이를 문자열 확인을 위한 Count로 사용합니다. 이로서 각 문자열의 길이가 같아야만 한다는 것을 어렴풋이 추측이 가능해집니다. EBX와 ECX를 4byte씩 가져온 뒤에 내용을 비교후 EDX를 감소시키는데 내용이 다르거나 EDX가 0이 되는 경우 함수를 바로 종료시키게 됩니다.

```
00403B57   8B1F          MOV EBX,DWORD PTR DS:[EDI]
00403B59   39D9          CMP ECX,EBX
00403B5B v 75 58         JNZ SHORT 05_UPXUn.00403BB5
00403B5D   4A            DEC EDX
00403B5E v 74 15         JE SHORT 05_UPXUn.00403B75
00403B60   8B4E 04       MOV ECX,DWORD PTR DS:[ESI+4]
00403B63   8B5F 04       MOV EBX,DWORD PTR DS:[EDI+4]
00403B66   39D9          CMP ECX,EBX
00403B68 v 75 4B         JNZ SHORT 05_UPXUn.00403BB5
00403B6A   83C6 08       ADD ESI,8
00403B6D   83C7 08       ADD EDI,8
00403B70   4A            DEC EDX
00403B71 ^ 75 E2         JNZ SHORT 05_UPXUn.00403B55
00403B73 v EB 06         JMP SHORT 05_UPXUn.00403B7B
```

다음 문자도 가져와 비교한 뒤에 ESI와 EDI를 각각 8씩 증가시켜 다시 위쪽으로 점프, 다시 연산을 실시합니다.

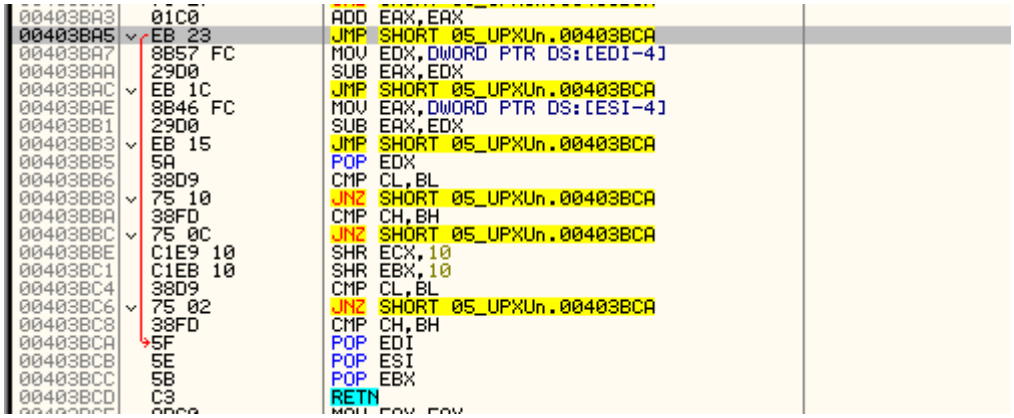
하지만 이번 루프에선 EDX가 3이었기에 절반만 실행 후 루프를 빠져나옵니다.

```
00403B73 v EB 06         JMP SHORT 05_UPXUn.00403B7B
00403B75   83C6 04       ADD ESI,4
00403B78   83C7 04       ADD EDI,4
00403B7B   5A            POP EDX
00403B7C   83E2 03       AND EDX,3
00403B7F v 74 22         JE SHORT 05_UPXUn.00403BA3
00403B81   8B0E          MOV ECX,DWORD PTR DS:[ESI]
00403B83   8B1F          MOV EBX,DWORD PTR DS:[EDI]
00403B85   38D9          CMP CL,BL
00403B87 v 75 41         JNZ SHORT 05_UPXUn.00403BCA
00403B89   4A            DEC EDX
00403B8A v 74 17         JE SHORT 05_UPXUn.00403BA3
00403B8C   38FD          CMP CH,BH
00403B8E v 75 3A         JNZ SHORT 05_UPXUn.00403BCA
00403B90   4A            DEC EDX
00403B91 v 74 10         JE SHORT 05_UPXUn.00403BA3
```

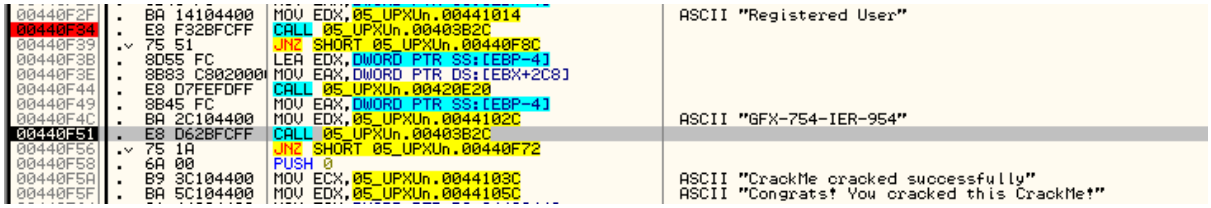
남게되는 3개의 문자 ser은 루프 밖에서 ESI/EDI + 4로 가져와 비교를 실시하며 CL, CH레지스터로 한글자 한글자씩 직접 가져와서 비교를 실시합니다.

```
00403B8E v 75 3A         JNZ SHORT 05_UPXUn.00403BCA
00403B90   4A            DEC EDX
00403B91 v 74 10         JE SHORT 05_UPXUn.00403BA3
00403B93   81E3 0000FF00 AND EBX,0FF0000
00403B99   81E1 0000FF00 AND ECX,0FF0000
00403B9F   39D9          CMP ECX,EBX
00403BA1 v 75 27         JNZ SHORT 05_UPXUn.00403BCA
00403BA3   01C0          ADD EAX,EAX
00403BA5 v EB 23         JMP SHORT 05_UPXUn.00403BCA
00403BA7   8B57 FC       MOV EDX,DWORD PTR DS:[EDI-4]
00403BAA   29D0          SUB EAX,EDX
00403BAC v EB 1C         JMP SHORT 05_UPXUn.00403BCA
00403BAE   8B46 FC       MOV EAX,DWORD PTR DS:[ESI-4]
00403BB1   29D0          SUB EAX,EDX
00403BB3 v EB 15         JMP SHORT 05_UPXUn.00403BCA
```

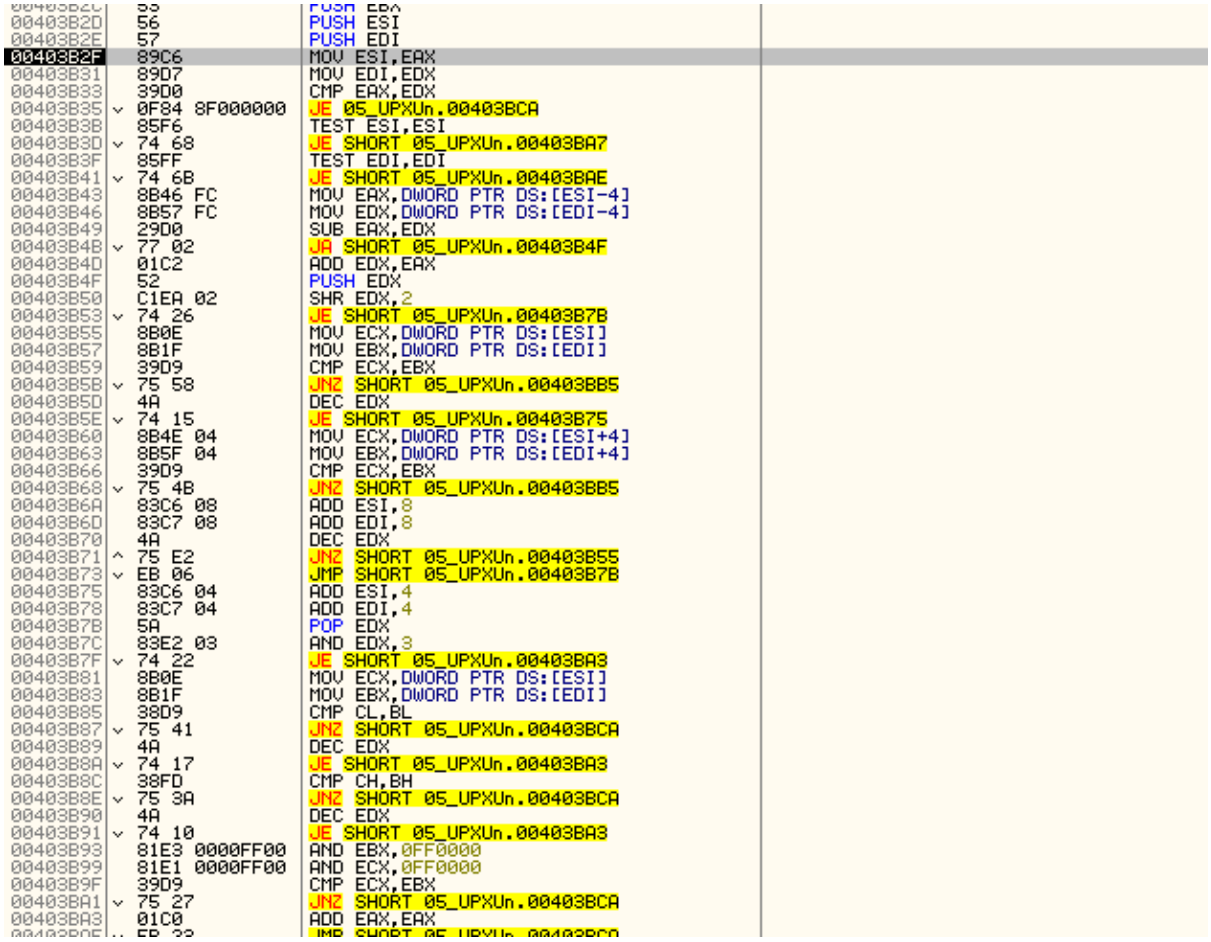
마지막 글자는 00FF0000으로 AND연산하여 1글자로 한 뒤 다시 비교를 실시합니다.



그후 EAX값을 2배로 하고 리턴하여 빠져 나옵니다.



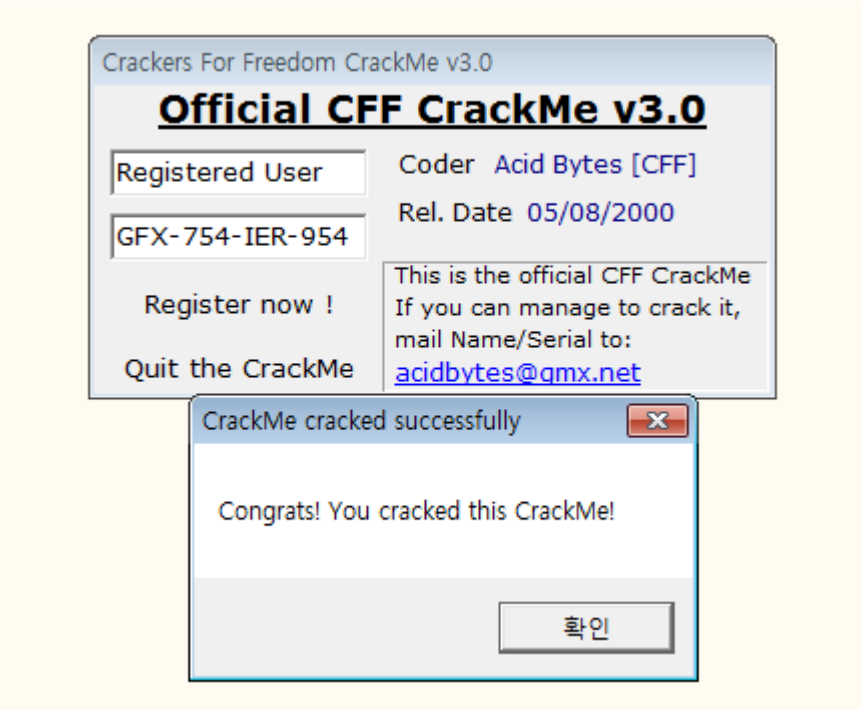
ZF값이 0인채로 빠져 나왔으므로 이후 JNZ에는 걸리지 않고 다음 함수를 호출합니다.



시리얼 값을 비교하는 함수구문입니다만 유저 이름을 비교하는 구문과 동일함을 볼 수 있습니다

이로서 유저네임과 시리얼값은 기본적으로 주어졌던

Registered User / GFX-754-IER-954임을 알 수가 있고 이를 입력하게 되면



이렇게 성공창이 뜨는것을 확인할 수 있습니다.