A photograph of a desk setup. In the foreground, a silver laptop with the Apple logo is visible. To its right, there are several pens and pencils, a small eraser, and a black mouse. A spiral-bound notebook is also partially visible. A semi-transparent white box with a diagonal line is overlaid on the right side of the image, containing the text.

올리디버거를 활용한 리버싱

올리디버거

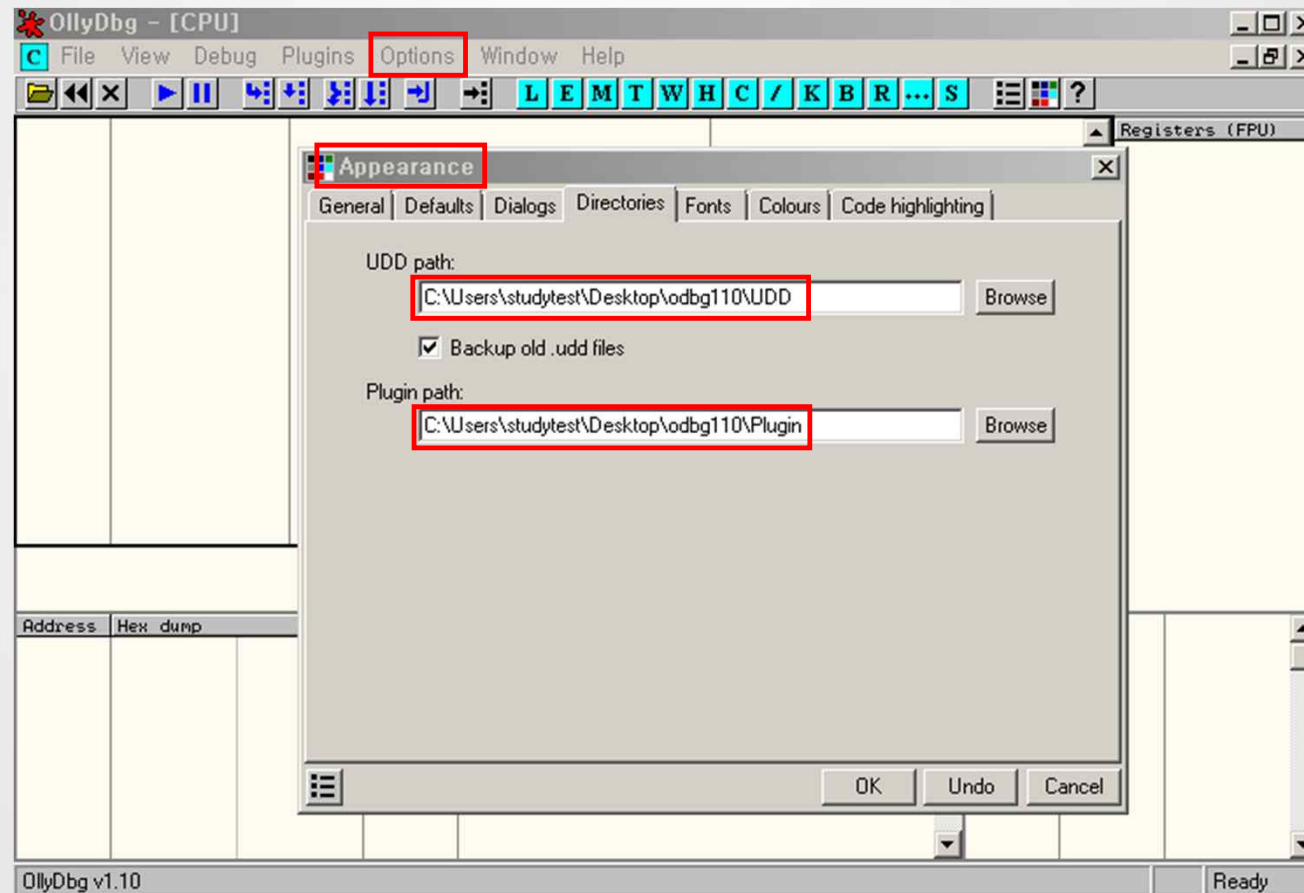
- 디스어셈블링
- 디버깅



- Version 1.10 is the final 1.x release.
- Version 2.0 is in development and is being written from the ground up.

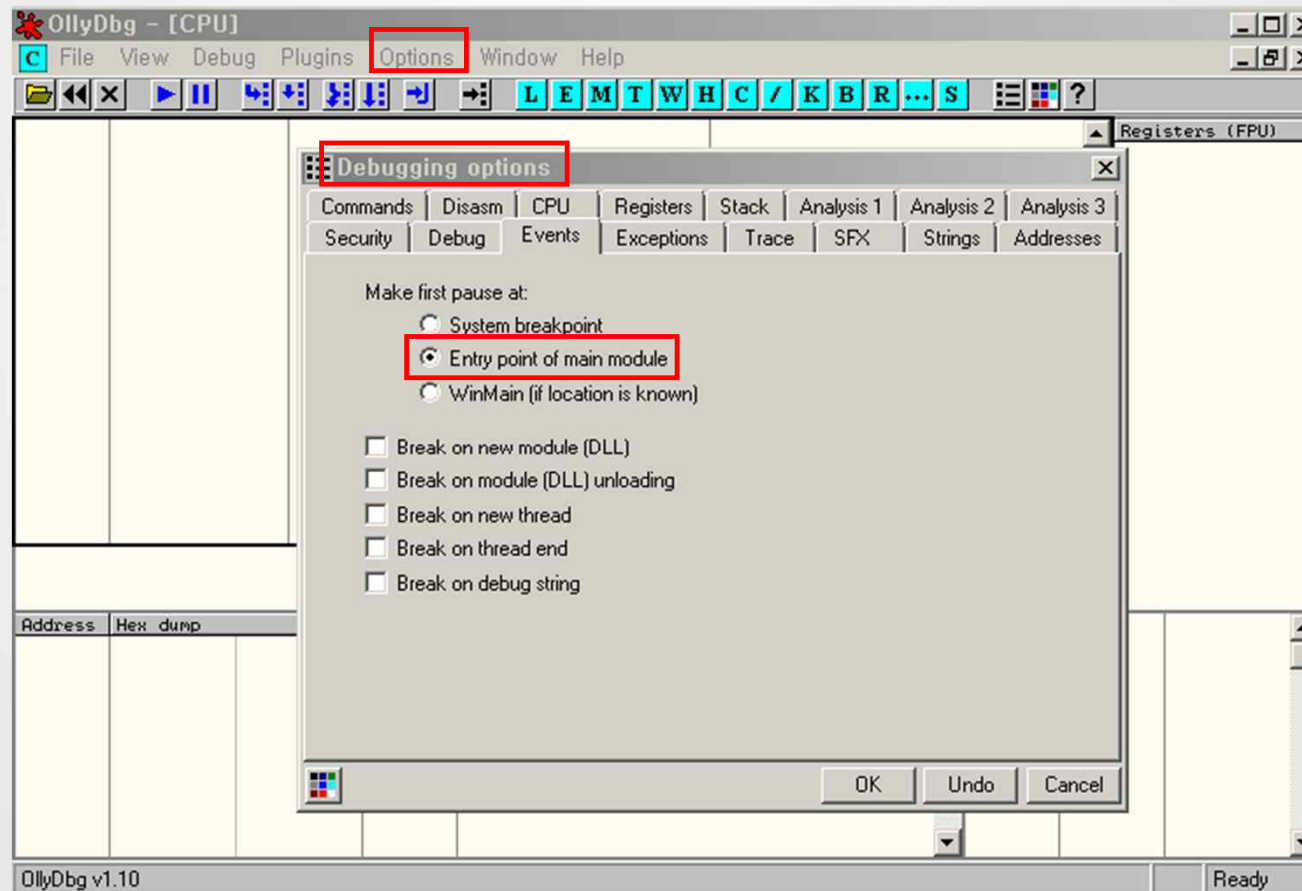
올리디버거 초기 설정 세팅

- UDD(백업 등 저장), 플러그인 디렉터리 경로 지정



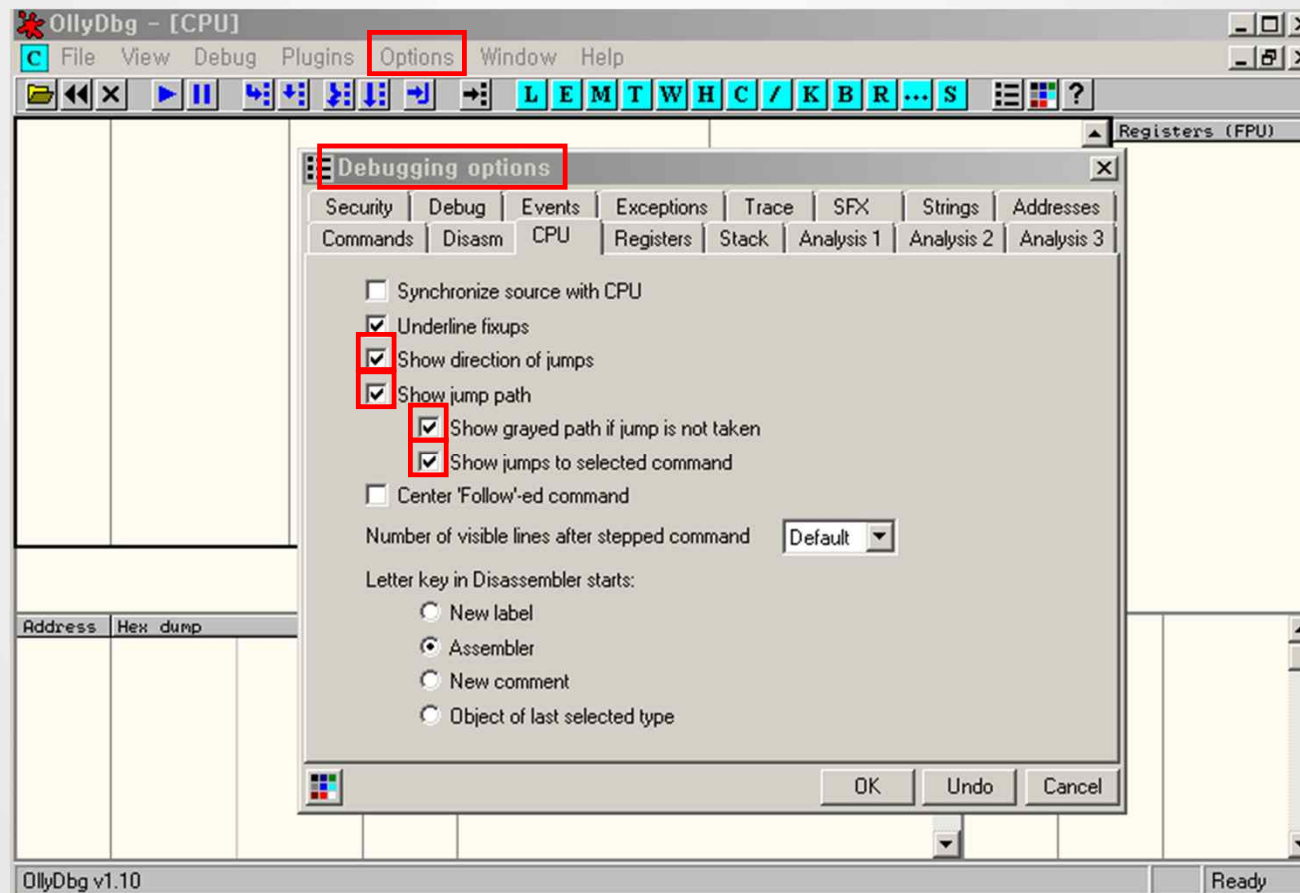
올리디버거 초기 설정 세팅

● Entry Point of main module 설정



올리디버거 초기 설정 세팅

●디버깅 중 점프 발생 시 경로가 화살표로 보이도록 표시 설정



올리디버거 화면

The screenshot shows the OllyDbg interface with the following components and annotations:

- 메모리 주소 (Memory Address):** Points to the address column in the CPU window.
- 기계어 코드 (Machine Code):** Points to the hex dump column in the CPU window.
- 어셈블리어 코드 (Assembly Code):** Points to the assembly instruction column in the CPU window.
- 주석 창 (Comment Window):** Points to the comment column in the CPU window.
- 레지스터 (Registers):** Points to the Registers (FPU) window on the right.
- 변수 값 출력 창 (Variable Value Output Window):** Points to the variable window at the bottom.
- HEX DUMP 창 (HEX DUMP Window):** Points to the hex dump window at the bottom left.
- 스택 (Stack):** Points to the stack window at the bottom right.

Registers (FPU) Data:

Register	Value
EAX	0010FCBC
ECX	0010FC88
EDX	774371B4 ntdll.KiFas
EBX	7FFDE000
ESP	0010FC54
EBP	0010FCCC
ESI	00000000
EDI	00000000
EIP	00A2DDF2 mineswee.00
C 0	ES 0023 32bit 0(FFF
P 1	CS 001B 32bit 0(FFF
A 0	SS 0023 32bit 0(FFF
Z 0	DS 0023 32bit 0(FFF
S 1	FS 003B 32bit 7FFDF
T 0	GS 0000 NULL

Stack Data:

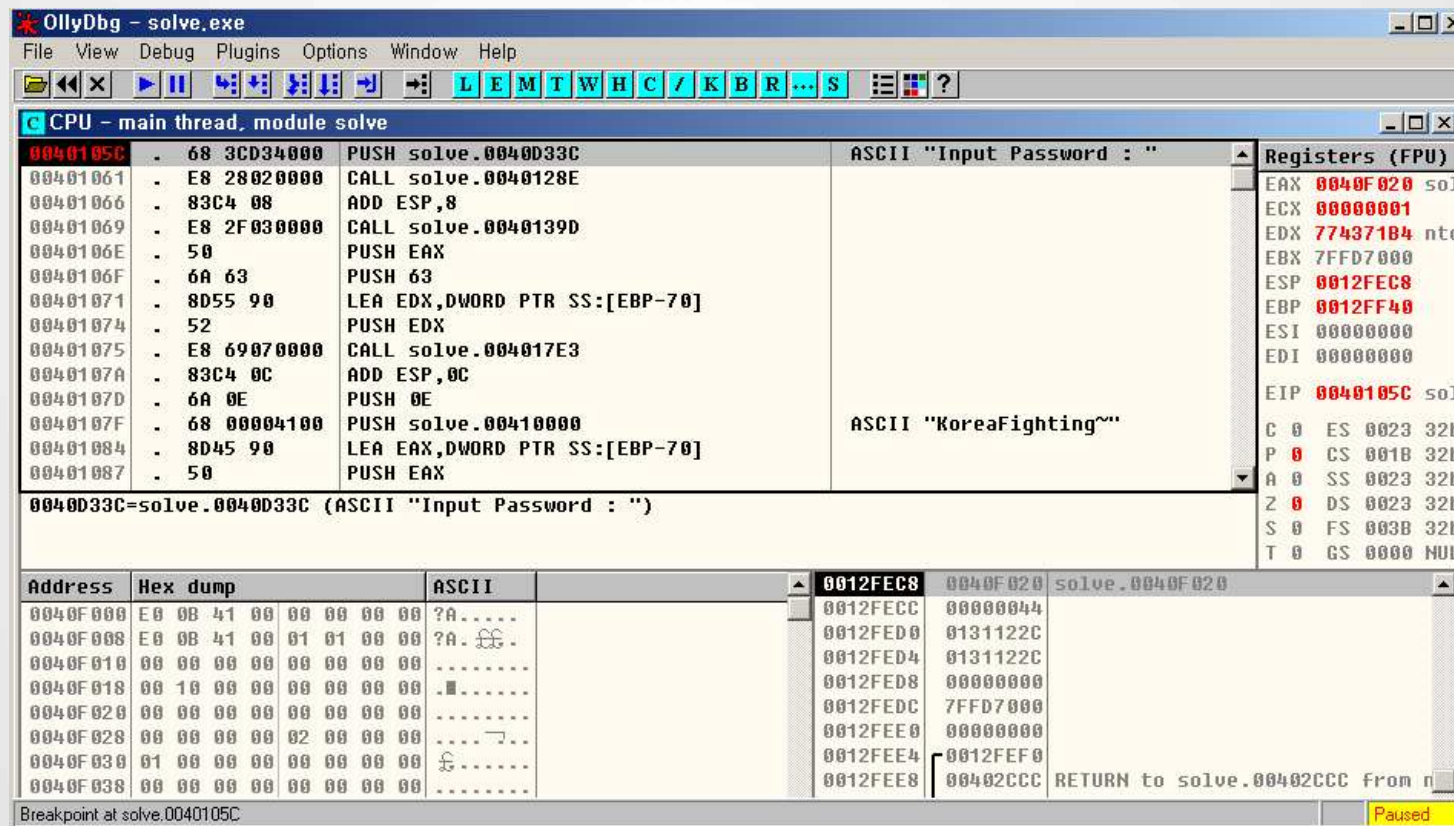
Address	Hex dump	ASCII
00A7E000	B8 5D A0 00 00 00 00 00	??....
00A7E008	2E 3F 41 56 42 6F 61 72	..?AVBoar
00A7E010	64 40 40 00 B8 5D A0 00	dg@.??
00A7E018	00 00 00 00 00 00 00 00
00A7E020	47 61 00 00 00 00 00 00	gAU
00A7E028	B8 5D A0 00 00 00 00 00	??....
00A7E030	2E 3F 41 56 47 61 6D 65	..?AVGame
00A7E038	53 74 61 74 40 40 00 00	Stat@@..
00A7E040	B8 5D A0 00 00 00 00 00	??....

올리디버거 주요 단축키

키	기능	설명
F2	Toggle	브레이크 포인트 지정/해제
F4	Execute till Cursor	커서 위치까지 실행
F9	Run	브레이크 포인트까지 실행
Ctrl+F2	Restart	처음부터 다시 재시작
F7	Step into	Call 함수나 반복적으로 수행하는 Rep 명령을 만났을 때 함수 내부로 추적 또는 Rep 조건을 만족할 때까지 계속 수행
F8	Step over	함수 내부로 추적하지 않고 함수 실행 후 그 다음 코드로 넘어감, Rep 같은 반복 명령도 한번에 처리하고 다음 코드로 실행
Ctrl+F9	Execute till Return	API 함수나 추적해 들어갈 필요 없는 함수 내부로 추적해 들어갔을 때 해당 함수의 ret 명령 지점까지 한번에 실행하고 함수를 나옴

올리디버거 기초분석방법

- 원하는 주소에서 브레이크포인트 지정(F2) 후 실행(F9)
- 이후 Step into(F7)이나 Step over(F8)로 명령 코드를 한 개씩 실행하며 분석



함수찾기

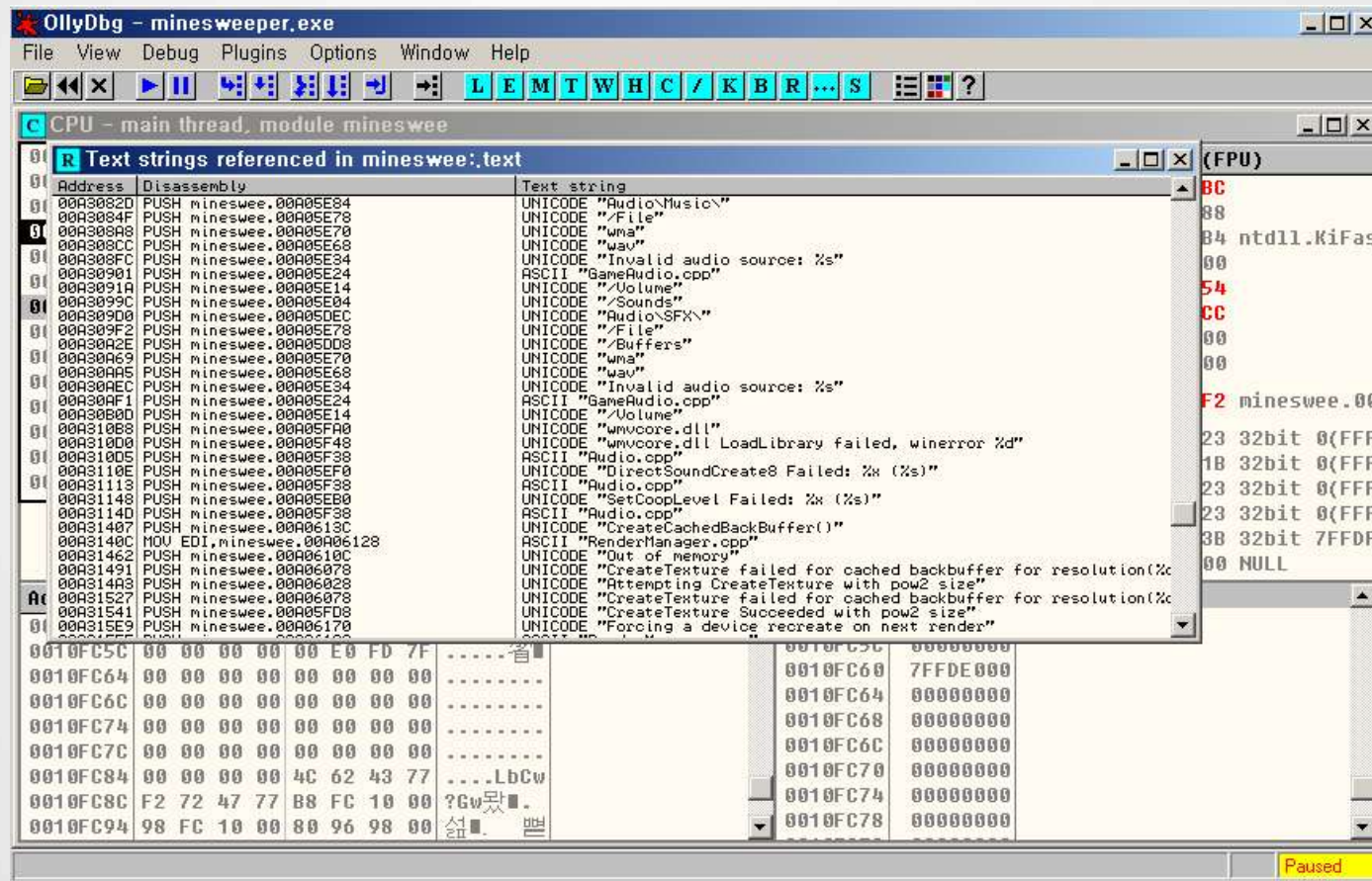
● All referenced text strings

The screenshot shows the OllyDbg interface for debugging minesweeper.exe. The CPU window displays assembly instructions for the main thread. A search menu is open, showing various search options, with 'All referenced text strings' selected.

Address	Hex dump	ASCII
0010FC54	41 71 AC 84 00 00 00 00	Aq...
0010FC5C	00 00 00 00 00 E0 FD 7F	...
0010FC64	00 00 00 00 00 00 00 00	...
0010FC6C	00 00 00 00 00 00 00 00	...
0010FC74	00 00 00 00 00 00 00 00	...
0010FC7C	00 00 00 00 00 00 00 00	...
0010FC84	00 00 00 00 4C 62 43 77	...L
0010FC8C	F2 72 47 77 B8 FC 10 00	?Gw...
0010FC94	98 FC 10 00 80 96 98 00	선...

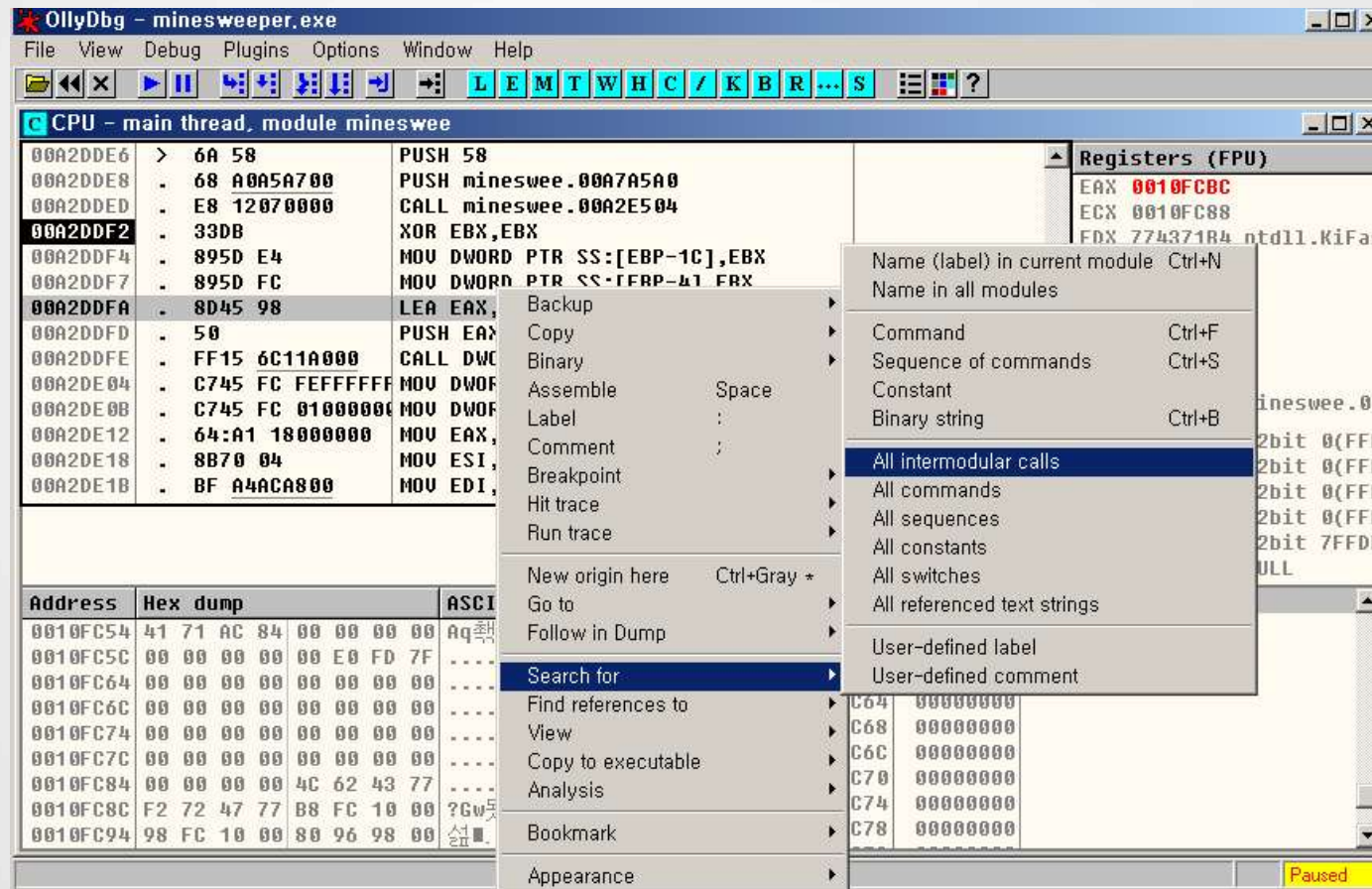
함수찾기

● All referenced text strings



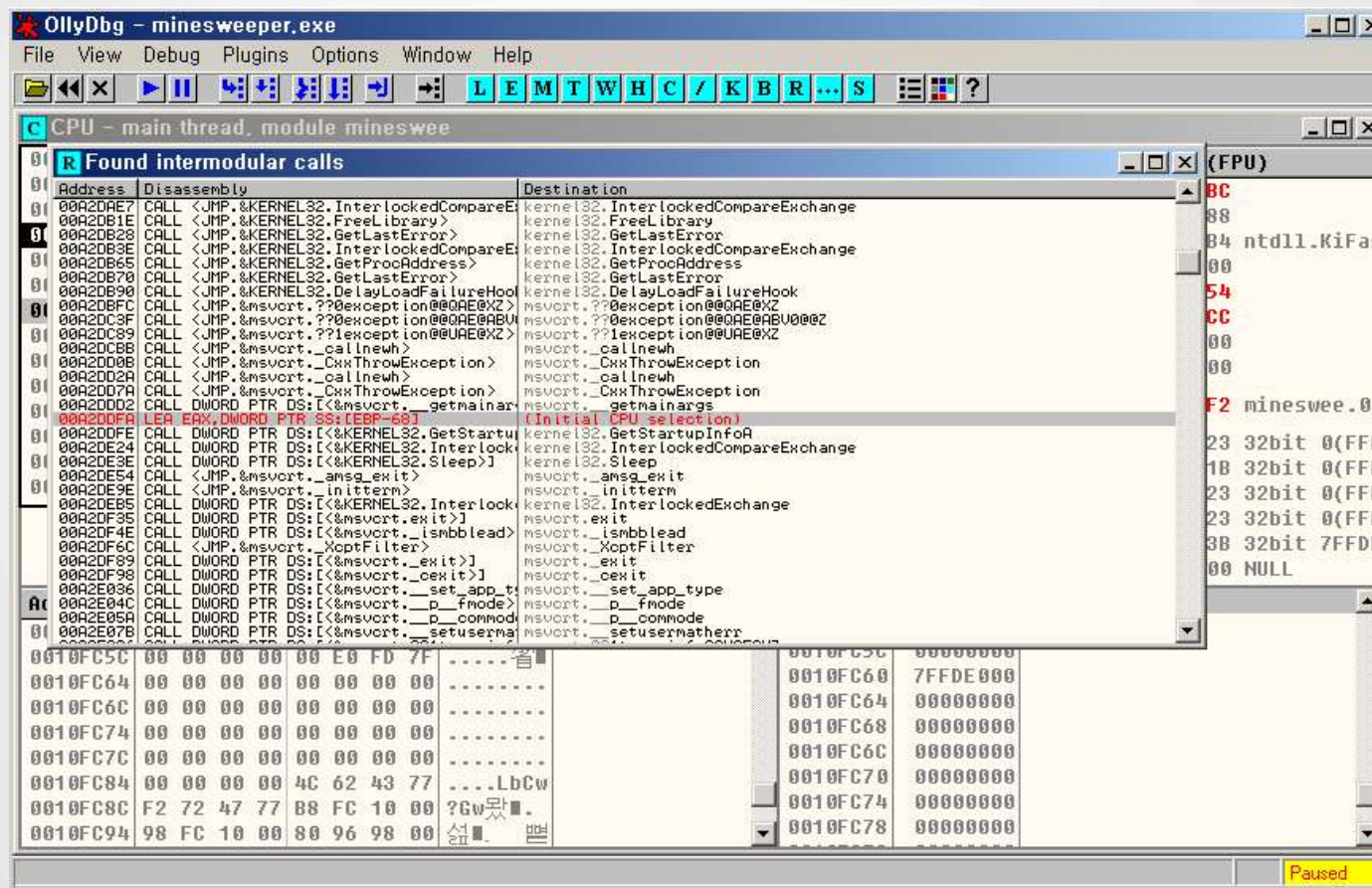
함수찾기

● All intermodule calls



함수찾기

● All intermodular calls



프로그램 변경 및 저장

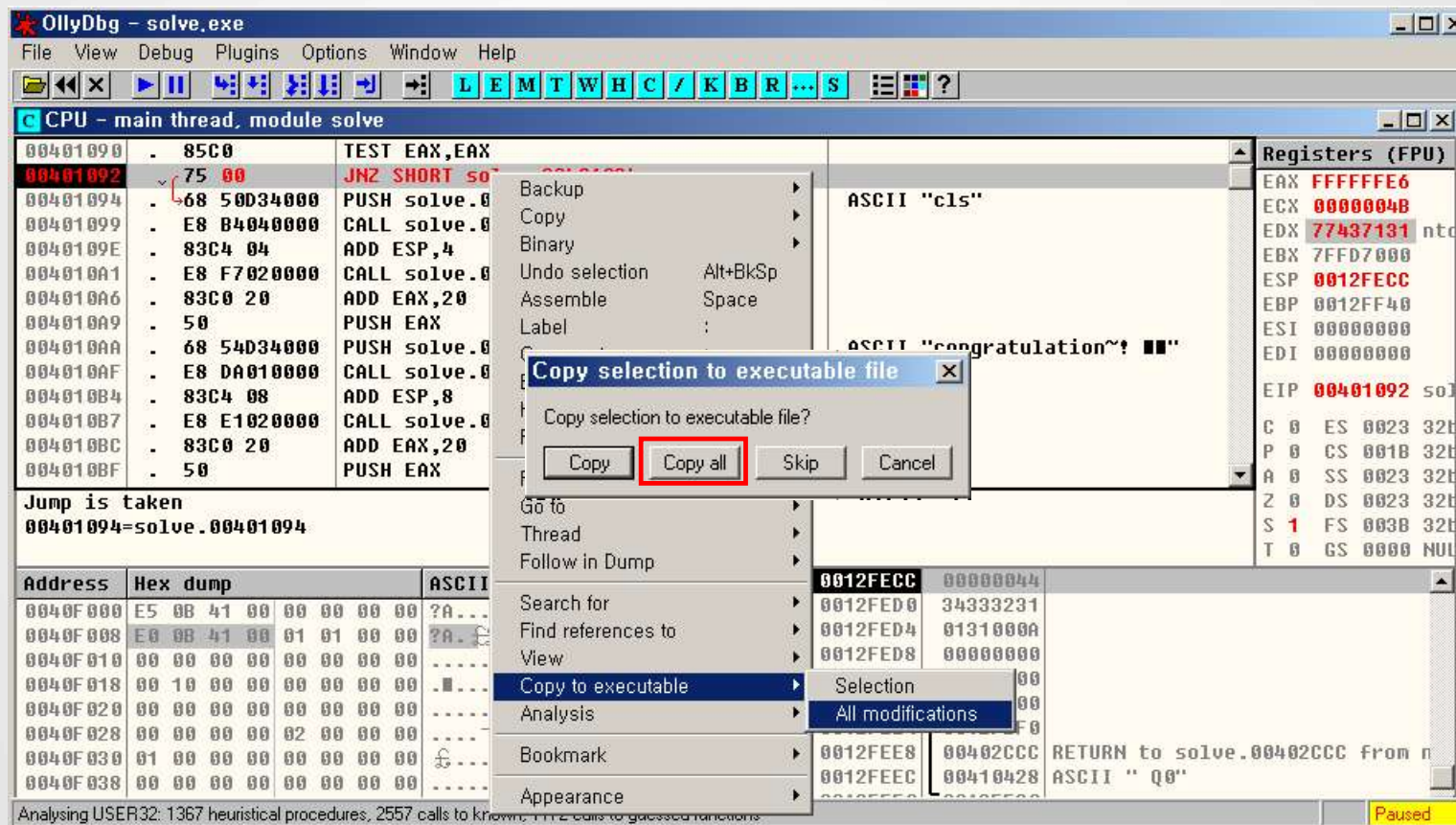
● 명령코드 수정

The screenshot shows the OllyDbg interface for the file 'solve.exe'. The CPU window displays assembly instructions. At address 00401092, the instruction is 'JNZ SHORT solve.00401094'. A dialog box titled 'Assemble at 00401092' is open, showing the selected instruction 'JNZ SHORT 00401094' and a checked option 'Fill with NOP's'. The registers window on the right shows the current state of CPU registers, with EIP pointing to 00401092. Below the assembly window, a memory dump is visible, showing hex and ASCII values for addresses 0040F000 to 0040F038.

Address	Hex dump	ASCII
0040F000	E5 0B 41 00 00 00 00 00	?A.....
0040F008	E0 0B 41 00 01 01 00 00	?A. GG.
0040F010	00 00 00 00 00 00 00 00
0040F018	00 10 00 00 00 00 00 00
0040F020	00 00 00 00 00 00 00 00
0040F028	00 00 00 00 02 00 00 00
0040F030	01 00 00 00 00 00 00 00
0040F038	00 00 00 00 00 00 00 00

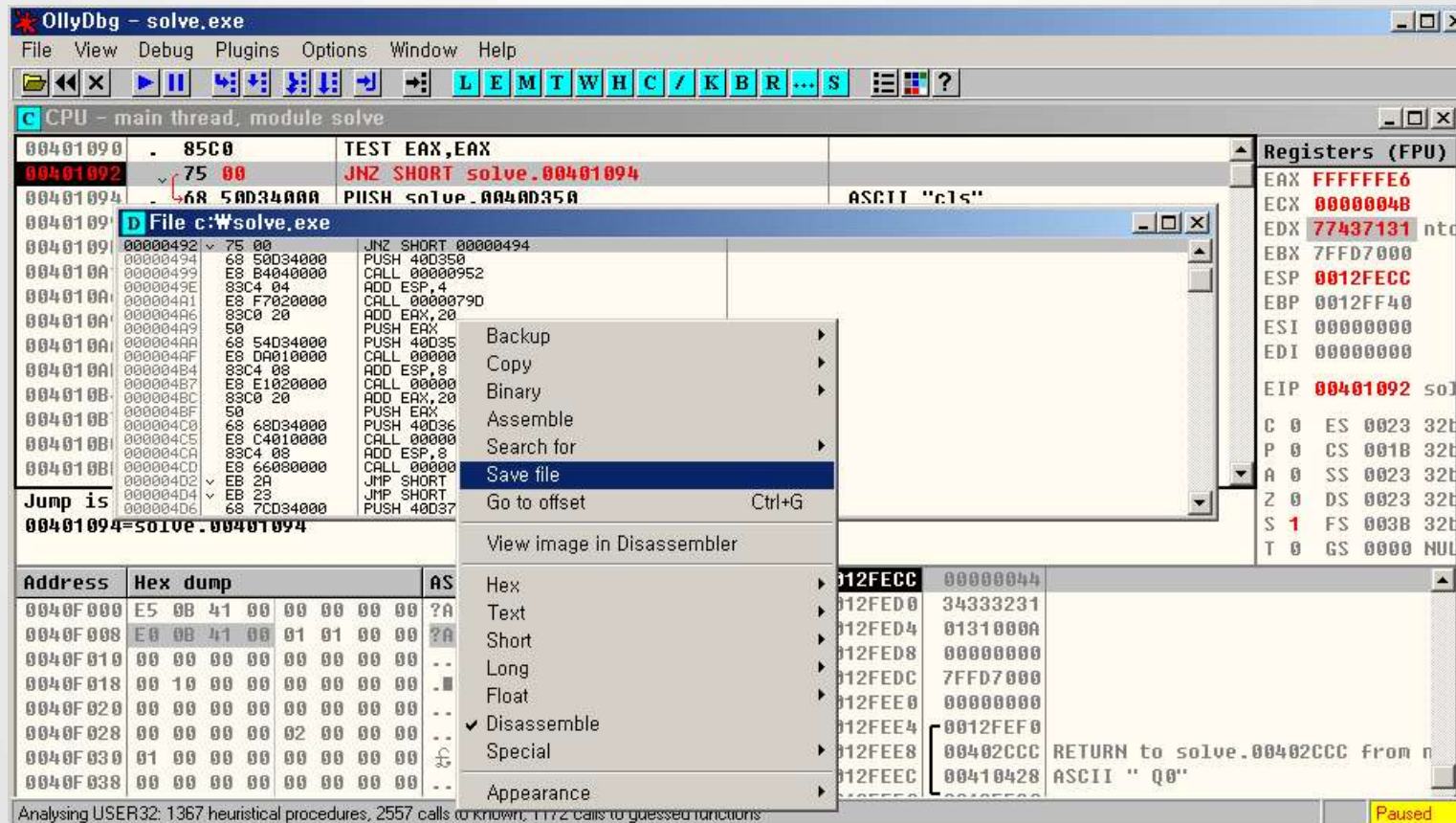
프로그램 변경 및 저장

● 프로그램 저장

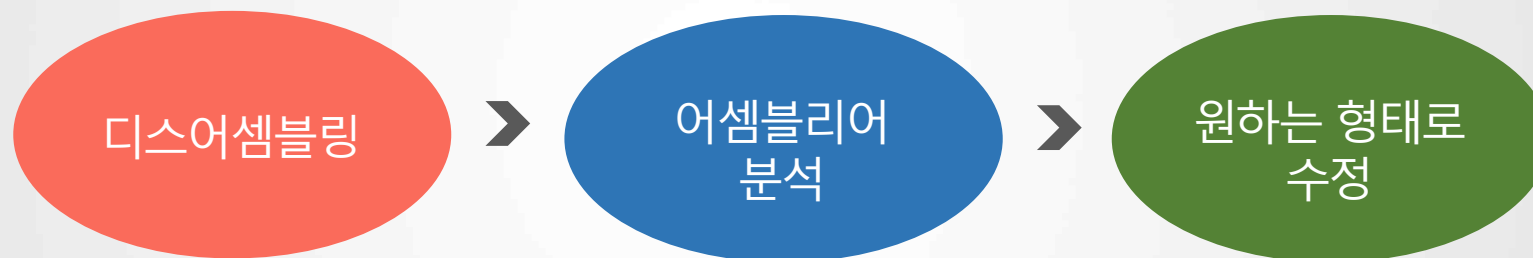


프로그램 변경 및 저장

● 프로그램 저장

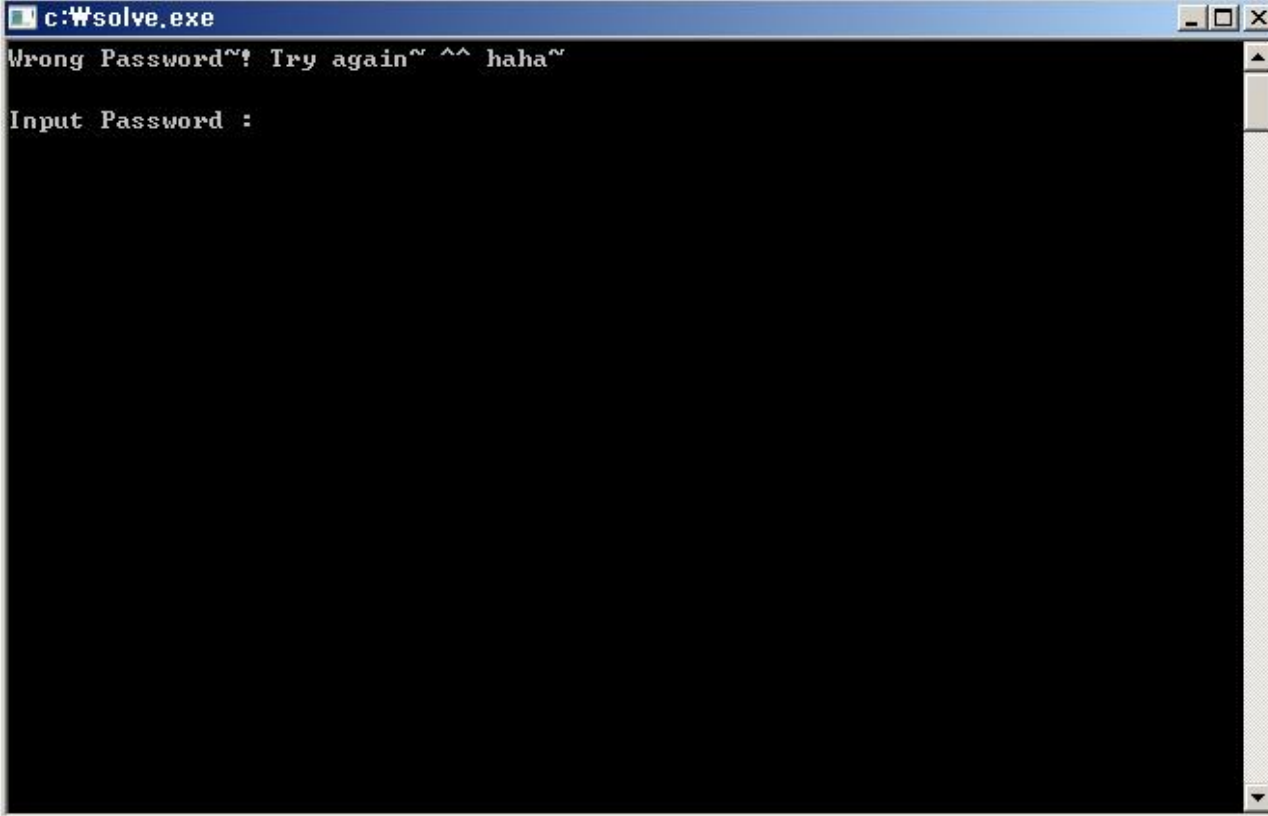


올리디버거를 통한 바이너리 파일 수정 단계



리버싱 문제 풀이(패스워드 찾기)

● 프로그램 실행 화면



```
c:\Wsolve.exe
Wrong Password~! Try again~ ^^ haha~
Input Password :
```

리버싱 문제 풀이(패스워드 찾기)

● All referenced text strings

The screenshot shows the OllyDbg interface for the file 'solve.exe'. The main window displays assembly code for the 'main thread, module solve'. The 'Text strings referenced in solve.exe' window is open, showing a list of text strings used in the program. The strings include 'Input Password:', 'KoreaFighting', 'Congratulations!', 'KoreaFighting~', and various error messages. The 'Registers (FPU)' window is also visible, showing the values of EAX, ECX, and EDX. A context menu is open over the 'Text strings' window, showing options like 'Follow in Disassembler', 'Search for text', and 'Copy to clipboard'.

Address	Disassembly	Text string
0040107F	PUSH solve.00401080	ASCII "KoreaFighting~"
00401084	LEA EAX,DWORD PTR SS:[EBP-70]	ASCII "Input Password:"
00401087	PUSH EAX	ASCII "KoreaFighting"
00401088	CALL solve.004011CE	(Initial CPU selection)
0040108D		ASCII "cls"
00401092		ASCII "Congratulations!"
00401094		ASCII "KoreaFighting~"
00401099		ASCII "cls"
0040109E		ASCII "cls"
004010A1		ASCII "cls"
004010A6		ASCII "cls"
004010A9		ASCII "cls"
004010AE		ASCII "cls"
004010B3		ASCII "cls"
004010B8		ASCII "cls"
004010BD		ASCII "cls"
004010C2		ASCII "cls"
004010C7		ASCII "cls"
004010CC		ASCII "cls"
004010D1		ASCII "cls"
004010D6		ASCII "cls"
004010DB		ASCII "cls"
004010E0		ASCII "cls"
004010E5		ASCII "cls"
004010EA		ASCII "cls"
004010EF		ASCII "cls"
004010F4		ASCII "cls"
004010F9		ASCII "cls"
004010FE		ASCII "cls"
00401103		ASCII "cls"
00401108		ASCII "cls"
0040110D		ASCII "cls"
00401112		ASCII "cls"
00401117		ASCII "cls"
0040111C		ASCII "cls"
00401121		ASCII "cls"
00401126		ASCII "cls"
0040112B		ASCII "cls"
00401130		ASCII "cls"
00401135		ASCII "cls"
0040113A		ASCII "cls"
0040113F		ASCII "cls"
00401144		ASCII "cls"
00401149		ASCII "cls"
0040114E		ASCII "cls"
00401153		ASCII "cls"
00401158		ASCII "cls"
0040115D		ASCII "cls"
00401162		ASCII "cls"
00401167		ASCII "cls"
0040116C		ASCII "cls"
00401171		ASCII "cls"
00401176		ASCII "cls"
0040117B		ASCII "cls"
00401180		ASCII "cls"
00401185		ASCII "cls"
0040118A		ASCII "cls"
0040118F		ASCII "cls"
00401194		ASCII "cls"
00401199		ASCII "cls"
0040119E		ASCII "cls"
004011A3		ASCII "cls"
004011A8		ASCII "cls"
004011AD		ASCII "cls"
004011B2		ASCII "cls"
004011B7		ASCII "cls"
004011BC		ASCII "cls"
004011C1		ASCII "cls"
004011C6		ASCII "cls"
004011CB		ASCII "cls"
004011D0		ASCII "cls"
004011D5		ASCII "cls"
004011DA		ASCII "cls"
004011DF		ASCII "cls"
004011E4		ASCII "cls"
004011E9		ASCII "cls"
004011EE		ASCII "cls"
004011F3		ASCII "cls"
004011F8		ASCII "cls"
004011FD		ASCII "cls"
00401202		ASCII "cls"
00401207		ASCII "cls"
0040120C		ASCII "cls"
00401211		ASCII "cls"
00401216		ASCII "cls"
0040121B		ASCII "cls"
00401220		ASCII "cls"
00401225		ASCII "cls"
0040122A		ASCII "cls"
0040122F		ASCII "cls"
00401234		ASCII "cls"
00401239		ASCII "cls"
0040123E		ASCII "cls"
00401243		ASCII "cls"
00401248		ASCII "cls"
0040124D		ASCII "cls"
00401252		ASCII "cls"
00401257		ASCII "cls"
0040125C		ASCII "cls"
00401261		ASCII "cls"
00401266		ASCII "cls"
0040126B		ASCII "cls"
00401270		ASCII "cls"
00401275		ASCII "cls"
0040127A		ASCII "cls"
0040127F		ASCII "cls"
00401284		ASCII "cls"
00401289		ASCII "cls"
0040128E		ASCII "cls"
00401293		ASCII "cls"
00401298		ASCII "cls"
0040129D		ASCII "cls"
004012A2		ASCII "cls"
004012A7		ASCII "cls"
004012AC		ASCII "cls"
004012B1		ASCII "cls"
004012B6		ASCII "cls"
004012BB		ASCII "cls"
004012C0		ASCII "cls"
004012C5		ASCII "cls"
004012CA		ASCII "cls"
004012CF		ASCII "cls"
004012D4		ASCII "cls"
004012D9		ASCII "cls"
004012DE		ASCII "cls"
004012E3		ASCII "cls"
004012E8		ASCII "cls"
004012ED		ASCII "cls"
004012F2		ASCII "cls"
004012F7		ASCII "cls"
004012FC		ASCII "cls"
00401301		ASCII "cls"
00401306		ASCII "cls"
0040130B		ASCII "cls"
00401310		ASCII "cls"
00401315		ASCII "cls"
0040131A		ASCII "cls"
0040131F		ASCII "cls"
00401324		ASCII "cls"
00401329		ASCII "cls"
0040132E		ASCII "cls"
00401333		ASCII "cls"
00401338		ASCII "cls"
0040133D		ASCII "cls"
00401342		ASCII "cls"
00401347		ASCII "cls"
0040134C		ASCII "cls"
00401351		ASCII "cls"
00401356		ASCII "cls"
0040135B		ASCII "cls"
00401360		ASCII "cls"
00401365		ASCII "cls"
0040136A		ASCII "cls"
0040136F		ASCII "cls"
00401374		ASCII "cls"
00401379		ASCII "cls"
0040137E		ASCII "cls"
00401383		ASCII "cls"
00401388		ASCII "cls"
0040138D		ASCII "cls"
00401392		ASCII "cls"
00401397		ASCII "cls"
0040139C		ASCII "cls"
004013A1		ASCII "cls"
004013A6		ASCII "cls"
004013AB		ASCII "cls"
004013B0		ASCII "cls"
004013B5		ASCII "cls"
004013BA		ASCII "cls"
004013BF		ASCII "cls"
004013C4		ASCII "cls"
004013C9		ASCII "cls"
004013CE		ASCII "cls"
004013D3		ASCII "cls"
004013D8		ASCII "cls"
004013DD		ASCII "cls"
004013E2		ASCII "cls"
004013E7		ASCII "cls"
004013EC		ASCII "cls"
004013F1		ASCII "cls"
004013F6		ASCII "cls"
004013FB		ASCII "cls"
00401400		ASCII "cls"
00401405		ASCII "cls"
0040140A		ASCII "cls"
0040140F		ASCII "cls"
00401414		ASCII "cls"
00401419		ASCII "cls"
0040141E		ASCII "cls"
00401423		ASCII "cls"
00401428		ASCII "cls"
0040142D		ASCII "cls"
00401432		ASCII "cls"
00401437		ASCII "cls"
0040143C		ASCII "cls"
00401441		ASCII "cls"
00401446		ASCII "cls"
0040144B		ASCII "cls"
00401450		ASCII "cls"
00401455		ASCII "cls"
0040145A		ASCII "cls"
0040145F		ASCII "cls"
00401464		ASCII "cls"
00401469		ASCII "cls"
0040146E		ASCII "cls"
00401473		ASCII "cls"
00401478		ASCII "cls"
0040147D		ASCII "cls"
00401482		ASCII "cls"
00401487		ASCII "cls"
0040148C		ASCII "cls"
00401491		ASCII "cls"
00401496		ASCII "cls"
0040149B		ASCII "cls"
004014A0		ASCII "cls"
004014A5		ASCII "cls"
004014AA		ASCII "cls"
004014AF		ASCII "cls"
004014B4		ASCII "cls"
004014B9		ASCII "cls"
004014BE		ASCII "cls"
004014C3		ASCII "cls"
004014C8		ASCII "cls"
004014CD		ASCII "cls"
004014D2		ASCII "cls"
004014D7		ASCII "cls"
004014DC		ASCII "cls"
004014E1		ASCII "cls"
004014E6		ASCII "cls"
004014EB		ASCII "cls"
004014F0		ASCII "cls"
004014F5		ASCII "cls"
004014FA		ASCII "cls"
004014FF		ASCII "cls"

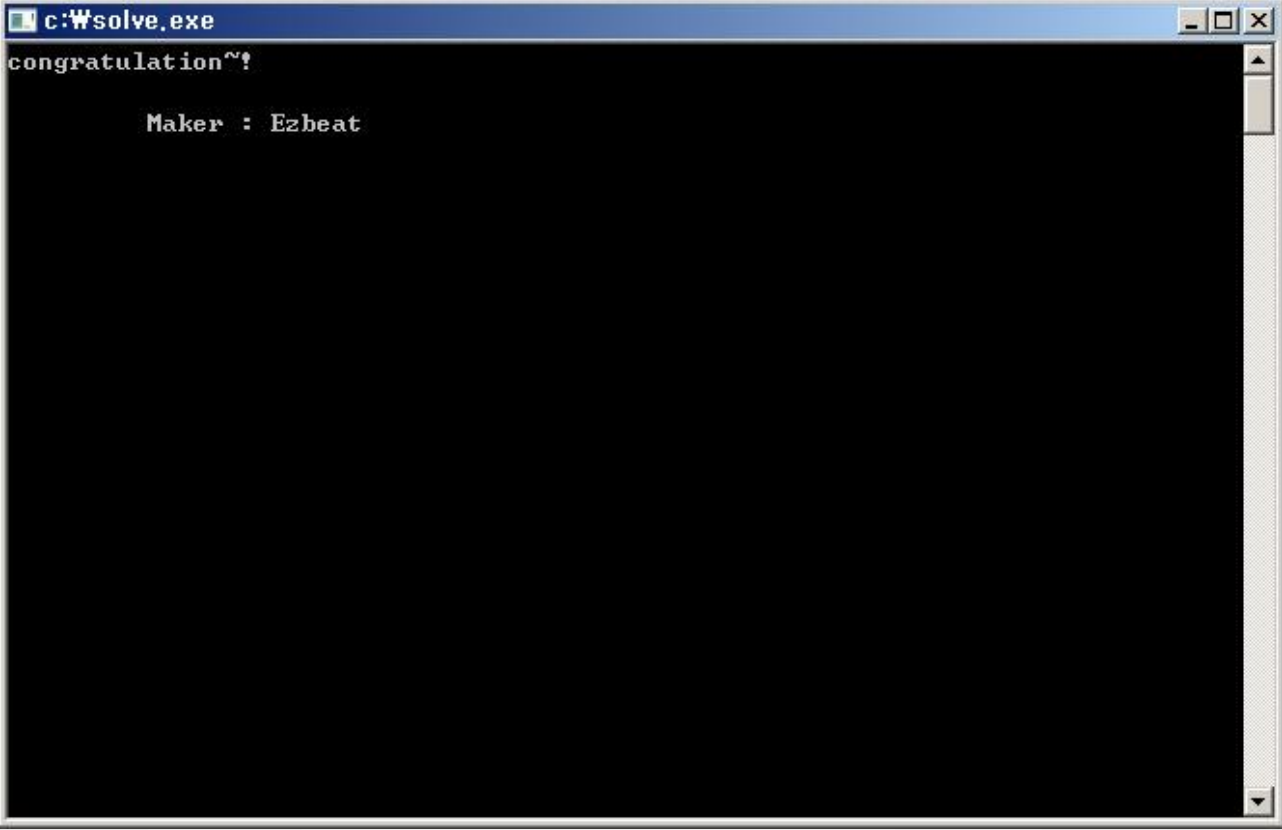
리버싱 문제 풀이(패스워드 찾기)

● 어셈블리어 수정

00401090	. 85C0	TEST EAX,EAX	
00401092	EB 00	JMP SHORT solve.00401094	
00401094	68 50D34000	PUSH solve.0040D350	ASCII "cls"
00401099	E8 B4040000	CALL solve.00401552	
0040109E	83C4 04	ADD ESP,4	
004010A1	E8 F7020000	CALL solve.0040139D	
004010A6	83C0 20	ADD EAX,20	
004010A9	50	PUSH EAX	
004010AA	68 54D34000	PUSH solve.0040D354	ASCII "congratulation~! ■■"
004010AF	E8 DA010000	CALL solve.0040128E	
004010B4	83C4 08	ADD ESP,8	
004010B7	E8 E1020000	CALL solve.0040139D	
004010BC	83C0 20	ADD EAX,20	
004010BF	50	PUSH EAX	
004010C0	68 68D34000	PUSH solve.0040D368	ASCII 09," Maker : "
004010C5	E8 C4010000	CALL solve.0040128E	
004010CA	83C4 08	ADD ESP,8	
004010CD	E8 66080000	CALL solve.00401938	
004010D2	EB 2A	JMP SHORT solve.004010FE	
004010D4	EB 23	JMP SHORT solve.004010F9	
004010D6	68 7CD34000	PUSH solve.0040D37C	ASCII "cls"
004010DB	E8 72040000	CALL solve.00401552	
004010E0	83C4 04	ADD ESP,4	
004010E3	E8 B5020000	CALL solve.0040139D	
004010E8	83C0 20	ADD EAX,20	
004010EB	50	PUSH EAX	
004010EB	50	PUSH EAX	

리버싱 문제 풀이(패스워드 찾기)

● 결과 확인



```
c:\Wsolve.exe
congratulation~!

Maker : Ezbeat
```