

Code Engn Basic 7

4.Z320

elttzero@gmail.com

Challenges : Basic 07

Author : abex

Korea :

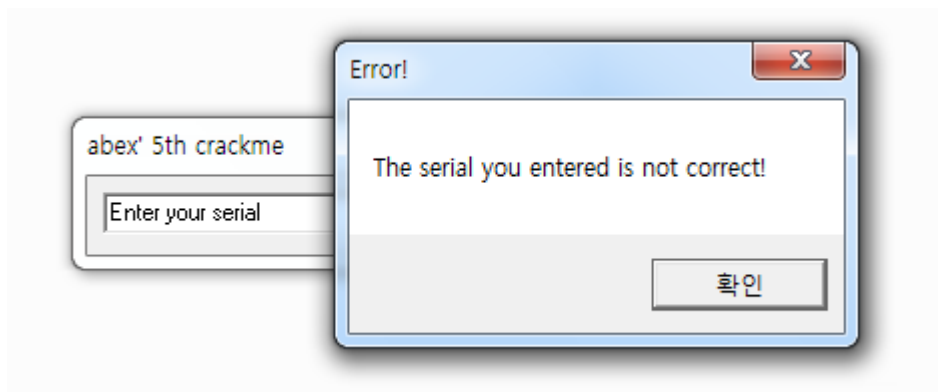
컴퓨터 C 드라이브의 이름이 CodeEngn 일경우 시리얼이 생성될때 CodeEngn은 "어떤것"으로 변경되는가

English :

Assuming the drive name of C is CodeEngn, what does CodeEngn transform into in the process of the serial construction

[Download](#)

7 번 문제는 C 드라이브 이름과 그에 따른 시리얼값을 찾아내는 문제입니다.



프로그램을 실행하면 키를 입력하는 창이 있으며 버튼을 누를 시 에러메세지가 뜰을 확인할 수 있습니다.

00401000	6A 00	PUSH 0	pModule = NULL
00401002	E8 34010000	CALL <JMP.&KERNEL32.GetModuleHandleA>	GetModuleHandleA
00401007	A3 EC234000	MOV DWORD PTR DS:[4023EC],EAX	
0040100C	6A 00	PUSH 0	
0040100E	68 29104000	PUSH 07.00401029	
00401013	6A 00	PUSH 0	
00401015	6A 01	PUSH 1	
00401017	FF35 EC234000	PUSH DWORD PTR DS:[4023EC]	
0040101D	E8 3D010000	CALL <JMP.&USER32.ShowDialogBoxParamA>	
00401022	6A 00	PUSH 0	
00401024	E8 1E010000	CALL <JMP.&KERNEL32.ExitProcess>	
00401029	C8 000000	ENTER 0,0	
00401030	017D 0C 1101	CMPL DWORD PTR DS:[40110C],EAX	

메인함수 부분이며 제대로된 함수 부분은 DialogBoxParamA 안에 있음을 알 수 있습니다.

00401069	. C2 1000	RETN 10	
0040106C	> 6A 25	PUSH 25	Count = 25 (37.)
0040106E	. 68 24234000	PUSH 07.00402324	Buffer = 07.00402324
00401073	. 6A 68	PUSH 68	ControlID = 68 (104.)
00401075	. FF75 08	PUSH DWORD PTR SS:[EBP+8]	hWnd
00401078	. E8 F4000000	CALL <JMP.&USER32.GetDlgItemTextA>	GetDlgItemTextA
0040107D	. 6A 00	PUSH 0	pFileSystemNameSize = NULL
0040107F	. 6A 00	PUSH 0	pFileSystemNameBuffer = NULL
00401081	. 68 C8204000	PUSH 07.004020C8	pFileSystemFlags = 07.004020C8
00401086	. 68 90214000	PUSH 07.00402190	pMaxFilenameLength = 07.00402190
00401088	. 68 94214000	PUSH 07.00402194	pVolumeSerialNumber = 07.00402194
00401090	. 6A 32	PUSH 32	MaxVolumeNameSize = 32 (50.)
00401092	. 68 5C224000	PUSH 07.0040225C	VolumeNameBuffer = 07.0040225C
00401097	. 6A 00	PUSH 0	RootPathName = NULL
00401099	. E8 B5000000	CALL <JMP.&KERNEL32.GetVolumeInformationA>	GetVolumeInformationA
0040109E	. 68 F3234000	PUSH 07.004023F3	StringToAdd = "4562-ABEX"
004010A3	. 68 5C224000	PUSH 07.0040225C	ConcatString = ""
004010A8	. E8 94000000	CALL <JMP.&KERNEL32.lstrcatA>	lstrcatA
004010AD	. B2 02	MOV DL,2	
004010AF	> 8305 5C224000	ADD DWORD PTR DS:[40225C],1	
004010B6	. 8305 5D224000	ADD DWORD PTR DS:[40225D],1	
004010BD	. 8305 5E224000	ADD DWORD PTR DS:[40225E],1	
004010C4	. 8305 5F224000	ADD DWORD PTR DS:[40225F],1	
004010CB	. FECA	DEC DL	
004010CD	. 75 E0	JNZ SHORT 07.004010AF	
004010CF	. 68 FD234000	PUSH 07.004023FD	StringToAdd = "L2C-5781"
004010D4	. 68 00204000	PUSH 07.00402000	ConcatString = ""
004010D9	. E8 63000000	CALL <JMP.&KERNEL32.lstrcatA>	lstrcatA

GetDlgItemTextA 로 사용자의 입력값을 받아들이며 GetVolumInformation 으로 사용자의 볼륨의 정보값을 읽어들이는 알 수 있습니다.

이때 볼륨네임은 0040225C 에 저장됨을 알 수 있으며 이는 lstrcat 에 의해 4562-ABEX와 합쳐짐을 알 수 있습니다.

00401088	. E8 94000000	CALL <JMP.&KERNEL32.lstrcatA>	lstrcatA
004010AD	. B2 02	MOV DL,2	
004010AF	> 8305 5C224000	ADD DWORD PTR DS:[40225C],1	
004010B6	. 8305 5D224000	ADD DWORD PTR DS:[40225D],1	
004010BD	. 8305 5E224000	ADD DWORD PTR DS:[40225E],1	
004010C4	. 8305 5F224000	ADD DWORD PTR DS:[40225F],1	
004010CB	. FECA	DEC DL	
004010CD	. 75 E0	JNZ SHORT 07.004010AF	
004010CF	. 68 FD234000	PUSH 07.004023FD	StringToAdd = "L2C-5781"
004010D4	. 68 00204000	PUSH 07.00402000	ConcatString = ""
004010D9	. E8 63000000	CALL <JMP.&KERNEL32.lstrcatA>	lstrcatA

이후 합쳐진 문자열 중 앞의 4byte 들은 1 씩 2 번 즉 2 씩 더해진 값이 되게 됩니다.

그리고 L2C-5781 은 빈 공백과 붙어지게 되어 하나의 문자열을 이룹니다.

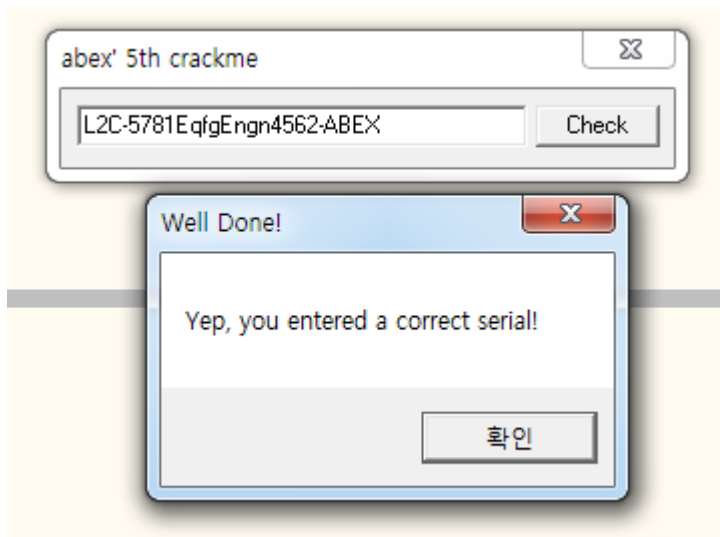
004010D4	. 68 00204000	PUSH 07.00402000	ConcatString = ""
004010D9	. E8 63000000	CALL <JMP.&KERNEL32.lstrcatA>	lstrcatA
004010DE	. 68 5C224000	PUSH 07.0040225C	StringToAdd = ""
004010E3	. 68 00204000	PUSH 07.00402000	ConcatString = ""
004010E8	. E8 54000000	CALL <JMP.&KERNEL32.lstrcatA>	lstrcatA
004010ED	. 68 24234000	PUSH 07.00402324	String2 = ""
004010F2	. 68 00204000	PUSH 07.00402000	String1 = ""
004010F7	. E8 51000000	CALL <JMP.&KERNEL32.lstrcmpiA>	lstrcmpiA
004010FC	. 83F8 00	CMP EAX,0	
004010FF	. 74 16	JE SHORT 07.00401117	
00401101	. 6A 00	PUSH 0	Style = MB_OK!MB_APPLMODAL
00401103	. 68 34244000	PUSH 07.00402434	Title = "Error!"
00401108	. 68 3B244000	PUSH 07.0040243B	Text = "The serial you entered is not correct!"
0040110D	. FF75 08	PUSH DWORD PTR SS:[EBP+8]	hOwner
00401110	. E8 56000000	CALL <JMP.&USER32.MessageBoxA>	MessageBoxA
00401115	. EB 16	JMP SHORT 07.0040112D	
00401117	> 6A 00	PUSH 0	Style = MB_OK!MB_APPLMODAL
00401119	. 68 06244000	PUSH 07.00402406	Title = "Well Done!"
0040111E	. 68 11244000	PUSH 07.00402411	Text = "Yep, you entered a correct serial!"
00401123	. FF75 08	PUSH DWORD PTR SS:[EBP+8]	hOwner
00401126	. E8 40000000	CALL <JMP.&USER32.MessageBoxA>	MessageBoxA
0040112B	. FR 00	JMP SHORT 07.0040112D	

그리고 뒤에 만들어진 문자열 00402000 과 앞서 만들어진 뒤 변경된 문자열은 합쳐지게 되어 L2C-5781XXXXXXXXX 가 되고 이후 사용자가 입력한 값과 비교, 결과에 따라 메시지를 띄우게 됩니다.

004010C0	75 E0	JNZ SHORT 07.004010AF	StringToAdd = "L2C-5781"
004010C1	68 F0234000	PUSH 07.004023FD	ConcatString = "L2C-5781EqfgEngn4562-ABEX"
004010C2	68 00204000	PUSH 07.00402000	lstrcatA
004010C3	E8 63000000	CALL <JMP.&KERNEL32.lstrcatA>	StringToAdd = "EqfgEngn4562-ABEX"
004010C4	68 5C224000	PUSH 07.0040225C	ConcatString = "L2C-5781EqfgEngn4562-ABEX"
004010C5	68 00204000	PUSH 07.00402000	lstrcatA
004010C6	E8 54000000	CALL <JMP.&KERNEL32.lstrcatA>	String2 = "Enter your serialaaa"
004010C7	68 24234000	PUSH 07.00402324	String1 = "L2C-5781EqfgEngn4562-ABEX"
004010C8	68 00204000	PUSH 07.00402000	lstrcatA
004010C9	E8 51000000	CALL <JMP.&KERNEL32.lstrcatA>	lstrcatA
004010CA	83F8 00	CMP EAX, 0	
004010CB	74 16	JE SHORT 07.00401117	
004010CC	6A 00	PUSH 0	Style = MB_OK!MB_APPLMODAL
004010CD	68 34244000	PUSH 07.00402434	Title = "Error!"
004010CE	68 3B244000	PUSH 07.0040243B	Text = "The serial you entered is not correct!"
004010CF	FF75 08	PUSH DWORD PTR SS:[EBP+8]	hOwner
004010D0	E8 56000000	CALL <JMP.&USER32.MessageBoxA>	MessageBoxA
004010D1	EB 16	JMP SHORT 07.00401120	
004010D2	6A 00	PUSH 0	Style = MB_OK!MB_APPLMODAL
004010D3	68 06244000	PUSH 07.00402406	Title = "Well Done!"
004010D4	68 11244000	PUSH 07.00402411	Text = "Yep, you entered a correct serial!"
004010D5	FF75 08	PUSH DWORD PTR SS:[EBP+8]	hOwner
004010D6	E8 40000000	CALL <JMP.&USER32.MessageBoxA>	MessageBoxA
004010D7	EB 00	JMP SHORT 07.00401120	
004010D8	6A 00	PUSH 0	Result = 0
004010D9	FF75 08	PUSH DWORD PTR SS:[EBP+8]	hWnd
004010DA	E8 22000000	CALL <JMP.&USER32.EndDialog>	EndDialog
004010DB	C9	LEAVE	
004010DC	C2 1000	RETN 10	
004010DD	FF25 6C304000	JMP DWORD PTR DS:[<&KERNEL32.GetModuleHandleA>]	kernel32.GetModuleHandleA
004010DE	FF25 70304000	JMP DWORD PTR DS:[<&KERNEL32.lstrcatA>]	kernel32.lstrcatA
004010DF	FF25 74304000	JMP DWORD PTR DS:[<&KERNEL32.ExitProcess>]	kernel32.ExitProcess
004010E0	FF25 78304000	JMP DWORD PTR DS:[<&KERNEL32.lstrcatA>]	kernel32.lstrcatA
004010E1	FF25 7C304000	JMP DWORD PTR DS:[<&KERNEL32.lstrcatA>]	kernel32.lstrcatA
004010E2	FF25 80304000	JMP DWORD PTR DS:[<&USER32.EndDialog>]	USER32.EndDialog
004010E3	FF25 84304000	JMP DWORD PTR DS:[<&USER32.EndDialog>]	USER32.EndDialog
004010E4	FF25 88304000	JMP DWORD PTR DS:[<&USER32.EndDialog>]	USER32.EndDialog

C 드라이브의 이름을 CodeEngn 으로 한 뒤에 프로그램을 동작시키면 드라이브 문자열의 앞 4byte 즉 Code 부분이 2 씩 더해져서 EqfgEngn 이 되고 이것이 이번 문제의 정답입니다.

이후 문자열이 합쳐져 L2C-5781EqfgEngn4562-ABEX 라는 문자열이 생성됩니다.



그리고 이것이 이 CrackME 의 시리얼이 됩니다.