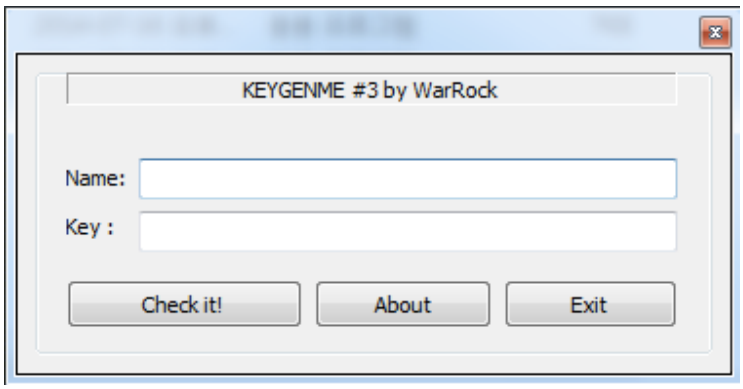
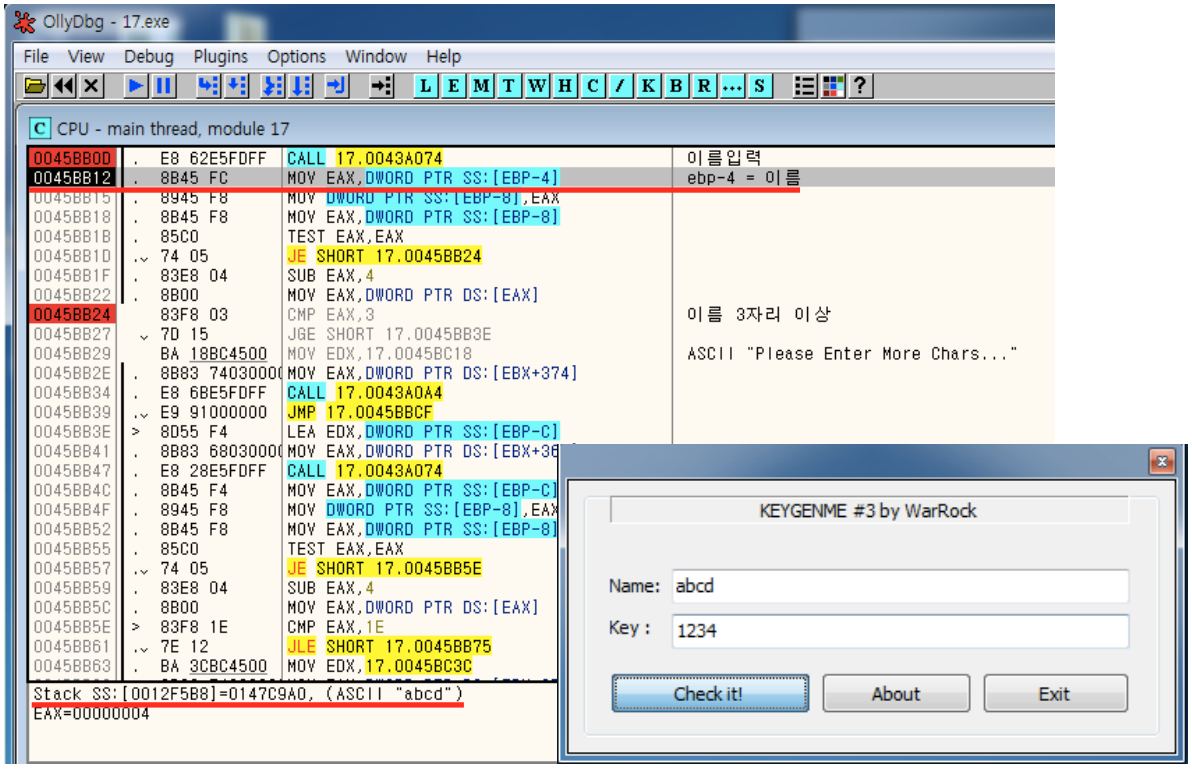
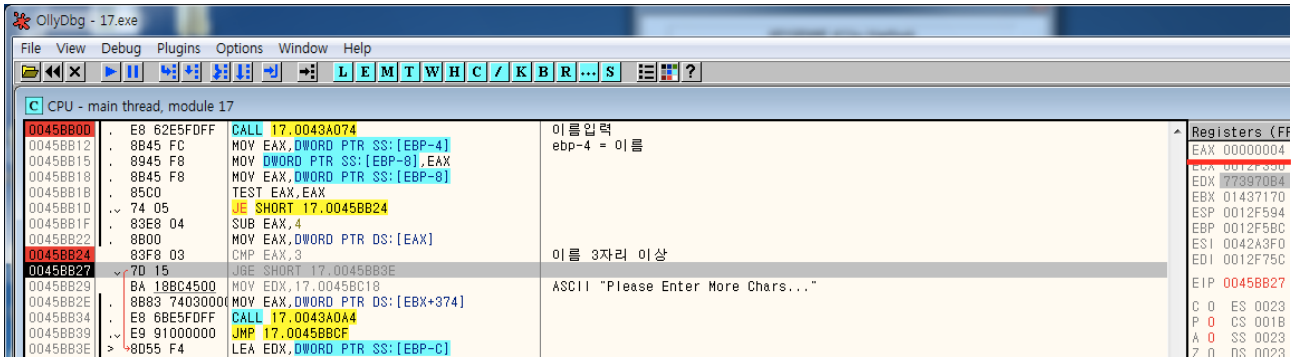


베이직 17번과 같은 문제입니다. 이름과 키를 입력받고 인증하는 프로그램입니다.

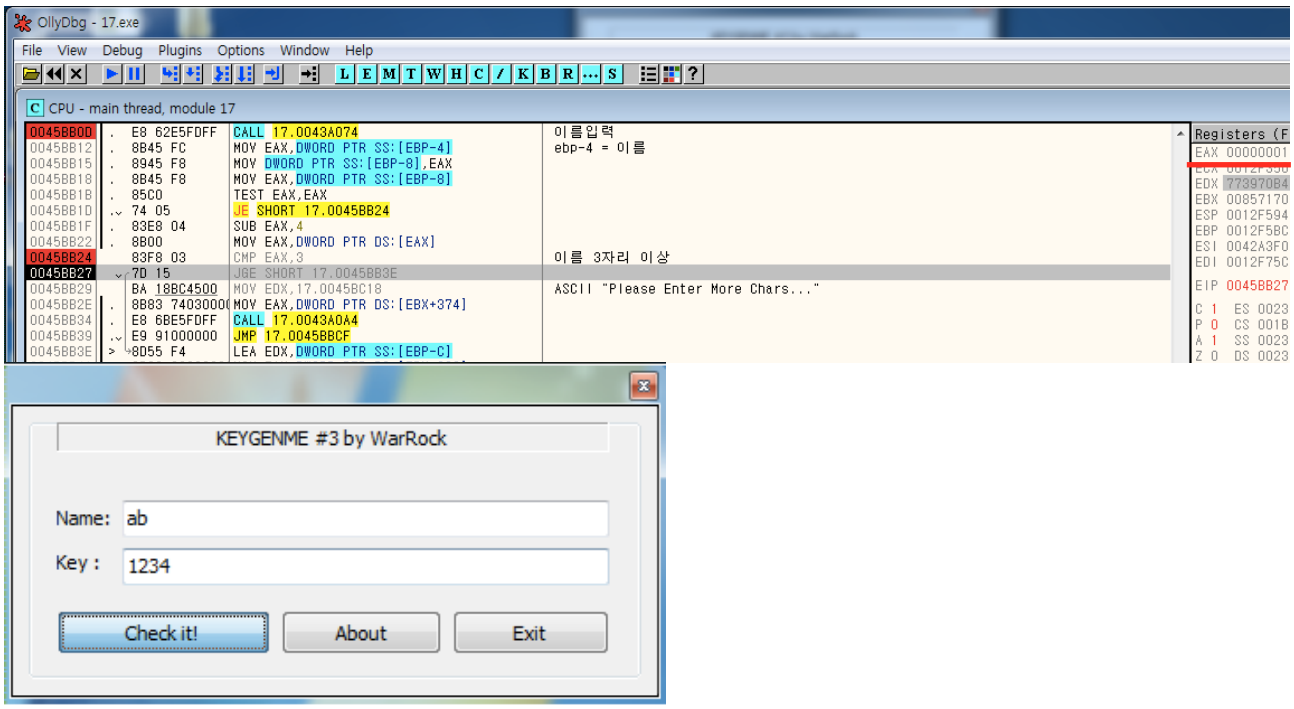


0045BB0D 주소의 함수가 이름을 불러오는 함수이고 ebp-4에 그 이름이 저장됨을 알 수 있습니다.

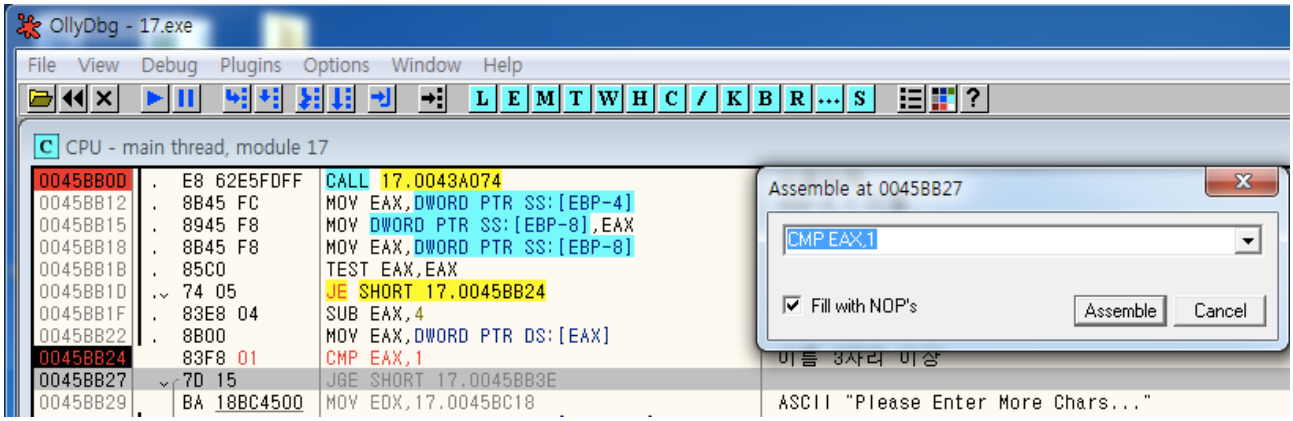


주석 부분을 보시면 이름의 글자수를 EAX에 받아와 비교합니다.

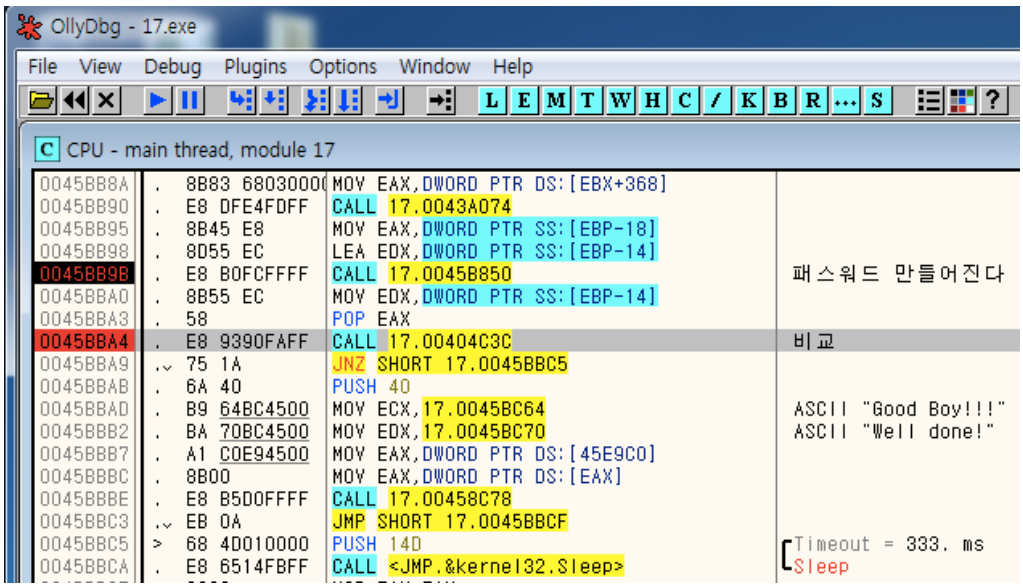
글자수가 3자리 이상이면 이 구문은 그냥 넘어가게 됩니다.



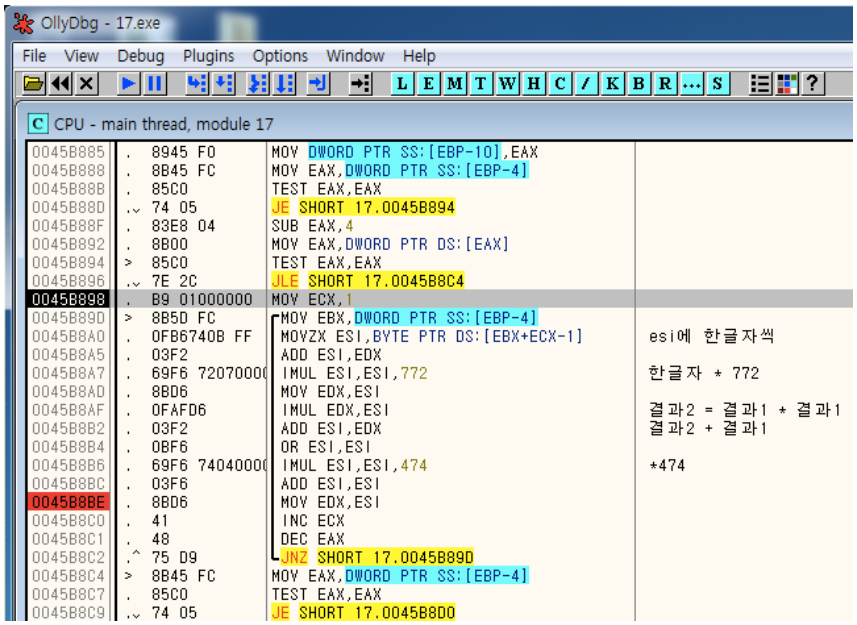
한 글자만 입력하면 EAX에 1이 들어가고 키 텍스트박스에는 이름을 더 길게 입력하라는 문구가 출력됩니다.



문제의 힌트에서 이름의 길이가 2이라고 했으니 비교구문의 3을 2로 수정합니다.



BreakPoint가 지정되어 있는 두 주소에서 패스워드가 만들어지고 입력한 패스워드와 비교하게 됩니다.

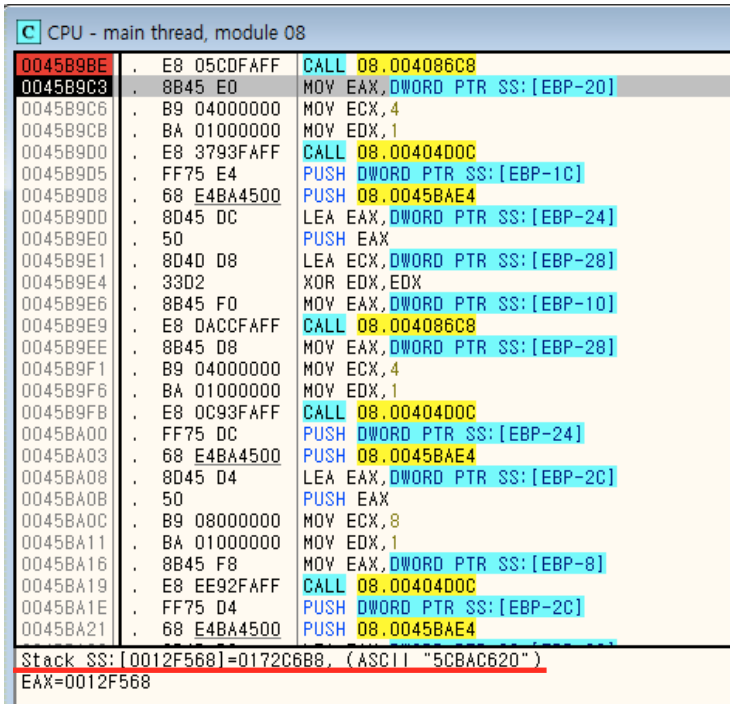


패스워드가 만들어지는 루틴입니다.

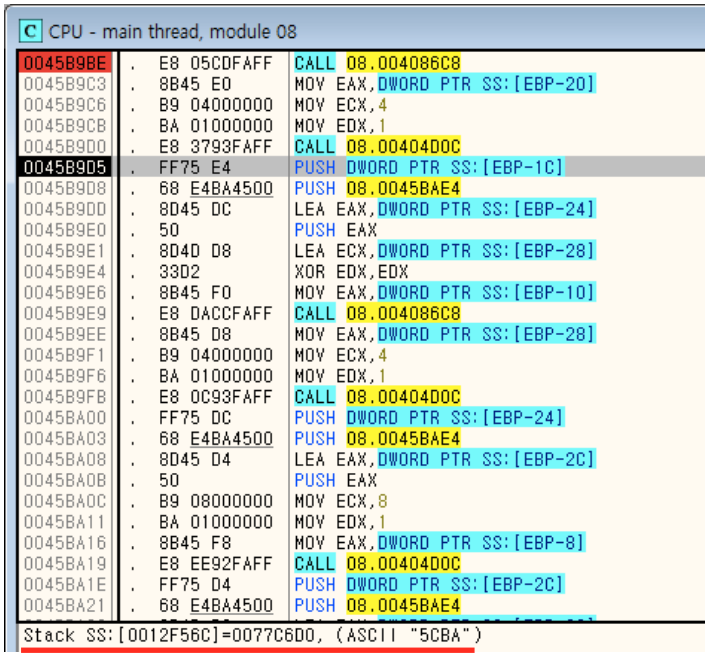
주석을 보면 알고리즘이 어떻게 이루어져 있는지 알 수 있습니다.

이 루틴은 이름 문자열의 길이만큼 반복합니다.

밑에 내려가면 더 많은 루틴이 있고 각각 특정 문자열을 만들어 냅니다.

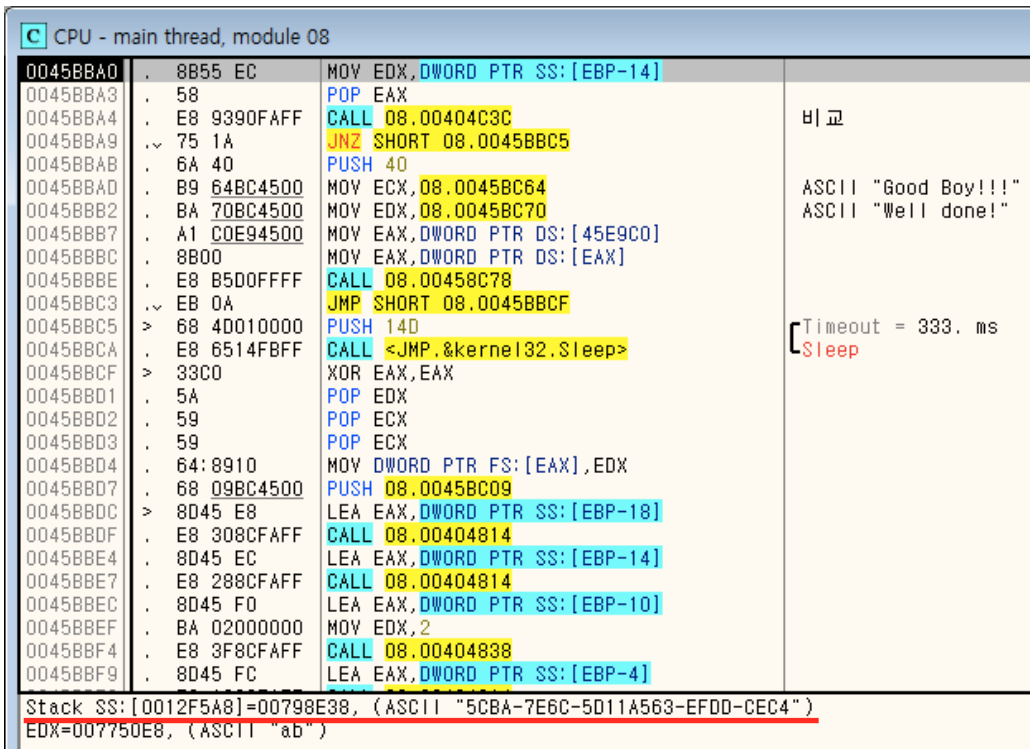


위에서 만들어진 문자열을 받아옵니다.



그리고 그 문자열 앞의 네자리를 뽑아냅니다.

이렇게 뽑아낸 글자들을 다 연결한 것이 최종 키가 됩니다.

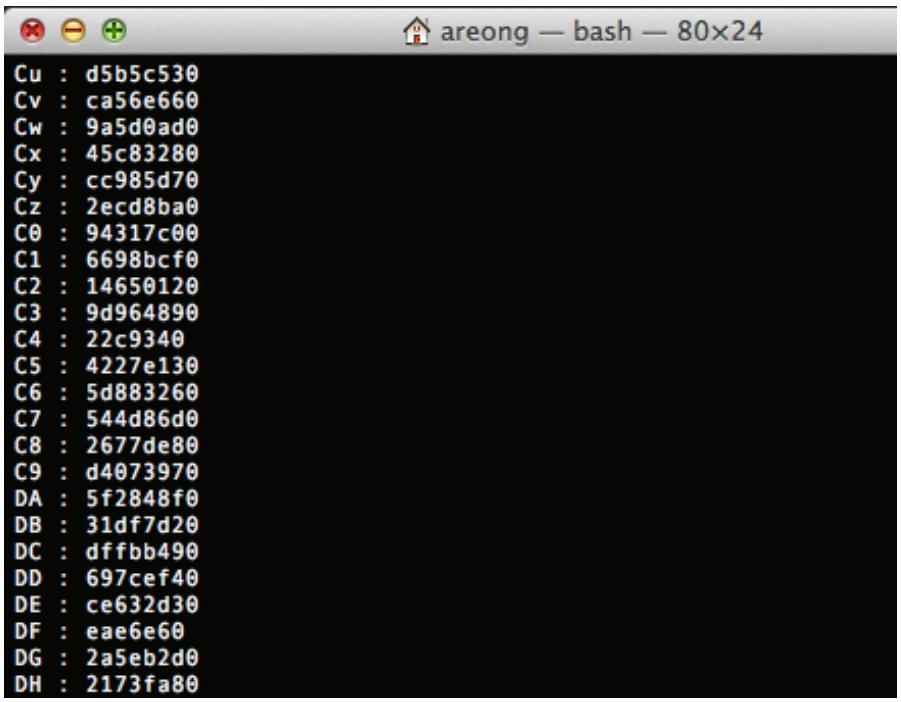


그림을 보시면 패스워드가 보이고 처음 4글자가 아까 뽑아낸 4글자인 것을 확인할 수 있습니다.

이름을 이용한 알고리즘으로 패스워드를 만들기 때문에 프로그래밍을 통해 원하고자 하는 값을 찾을 수 있습니다.

```
0x39, 0x77,
0x39, 0x78,
0x39, 0x79,
0x39, 0x7a,
0x39, 0x30,
0x39, 0x31,
0x39, 0x32,
0x39, 0x33,
0x39, 0x34,
0x39, 0x35,
0x39, 0x36,
0x39, 0x37,
0x39, 0x38,
0x39, 0x39,
};

    int esi,edx,i,j;

    //b=esi
    //c=edx


    for(i=0;i<1452;i++)
    {
        for(j=0;j<2;j++)
        {
            esi=a[i][j];
            esi=esi+edx;
            esi=esi*0x772;
            edx=esi;
            edx=edx*esi;
            esi=esi+edx;
            esi=esi*0x474;
            esi=esi+esi;
            edx=esi;
        }
        printf("%c%c : %x\n",a[i][0],a[i][1],edx);
        edx=0;

    }

    return 0;
}
```

조건에 만족되는 모든 HEX값을 알고리즘에 적용시키는 프로그램입니다.



이렇게 알고리즘이 적용된 글자들을 확인할 수 있고 조건에 맞는 문자를 MD5한것이 정답입니다.