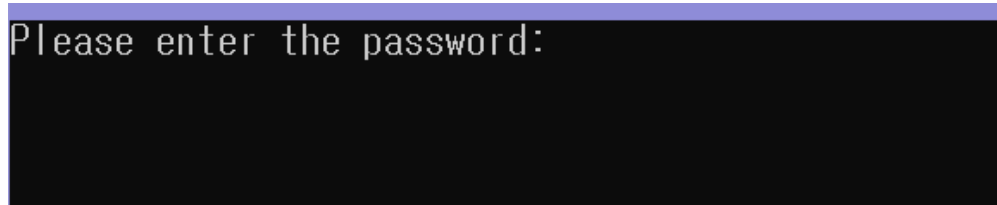


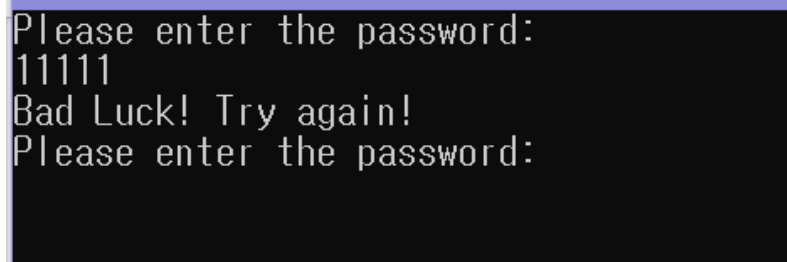
CodeEngn Basic RCE L13

: 정답은 무엇인가



```
Please enter the password:
```

exe 파일을 실행시키면 password를 입력하라는 콘솔 창이 나온다.



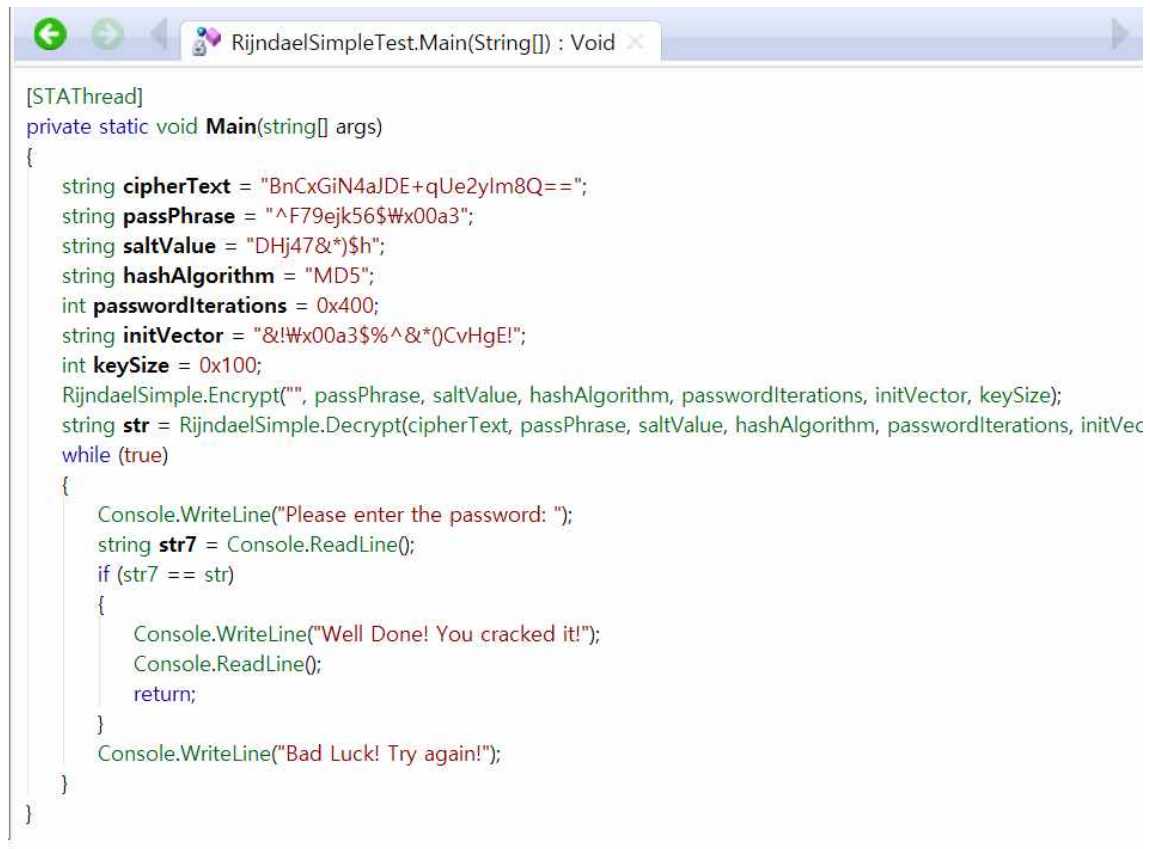
```
Please enter the password:
11111
Bad Luck! Try again!
Please enter the password:
```

아무거나 입력하면 Bad Luck! Try again! 이라는 스트링이 출력되고 다시 password를 입력하라는 창이 나온다.

```
int main() {
    int num; //사용자가 입력하는 값 (숫자로 가정)
    int key; // password값이 숫자라 가정
    key = XXXXXX // 키 값
    while(true){
        cout<<"Please enter the password"<<endl;
        cin>>num;
        if(num==key) {
            cout<<"성공메세지"<<endl;
            return false;
        }
        else
            cout<<"Bad Luck! Try again!"<<endl;
    }
}
```

exe 파일 프로그래밍이 이런식으로 이루어진 것이 아닐까 해서 한번 직접 코딩해 보았다!
프로그램 상 패스워드를 key라고 두었고, int로 변수 선언을 해주었다. cin으로 입력 받아서
그 값이 같으면 성공메세지, 아니라면 else로 에러메시지를 출력하도록 한다.

올리디버거에 올려서 분석해보려고 하니 오류가 발생한다. 검색해보니 .NET reflector 도구를 이용해야 한다고 해서 다운받아 주었다.



```
[STAThread]
private static void Main(string[] args)
{
    string cipherText = "BnCxCiN4aJDE+qUe2yIm8Q==";
    string passPhrase = "^F79ejk56$Wx00a3";
    string saltValue = "DHj47&*)$h";
    string hashAlgorithm = "MD5";
    int passwordIterations = 0x400;
    string initVector = "&!Wx00a3$%^&*)CvHgE!";
    int keySize = 0x100;
    RijndaelSimple.Encrypt("", passPhrase, saltValue, hashAlgorithm, passwordIterations, initVector, keySize);
    string str = RijndaelSimple.Decrypt(cipherText, passPhrase, saltValue, hashAlgorithm, passwordIterations, initVec
    while (true)
    {
        Console.WriteLine("Please enter the password: ");
        string str7 = Console.ReadLine();
        if (str7 == str)
        {
            Console.WriteLine("Well Done! You cracked it!");
            Console.ReadLine();
            return;
        }
        Console.WriteLine("Bad Luck! Try again!");
    }
}
```

.ConsoleApplication3 -> ConsoleApplication3.exe -> References -> {} - -> RijndaelSimpleTest -> Derived Types -> Main(String[]) : Void 로 들어가면 C# 코드가 나온다.

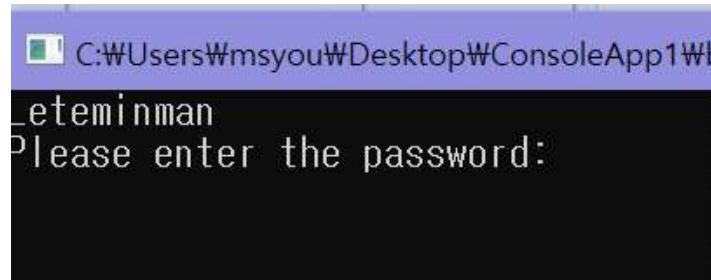
코드를 분석해보면 str7과 str이 같아야 성공메세지를 출력하고, 코드 상으로 str이 답인것을 알 수가 있다. 근데 해당 값이 해시알고리즘 MD5로 암호화 되어 있기 때문에 이를 복호화해서 푸는 것보다는 str을 출력하도록 프로그램을 수정해서 실행시켜보는 것이 더 효율적일 것이다. 따라서 while문 이전에 str을 출력하도록 바꾸어 주는데, 나는 비주얼 스튜디오가 있어서 이 툴을 사용해 수정했다. 원래 비주얼 스튜디오를 c++언어로만 사용해서 c#을 사용할 수 있도록 업데이트 해주었고,

```
Console.WriteLine("{0}", plainText);|
```

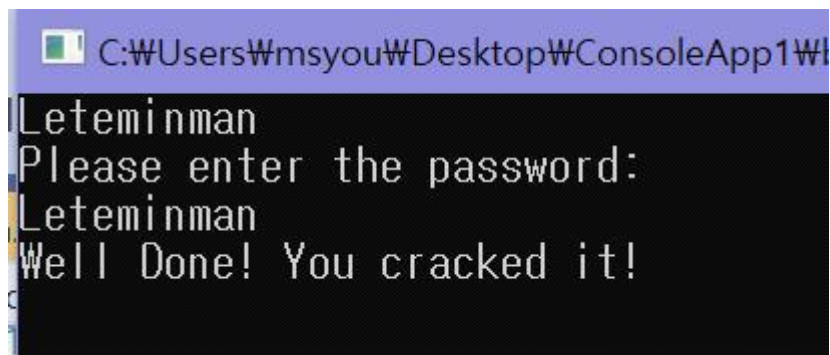
다음과 같은 코드를 추가한다.

비주얼 스튜디오로 어떻게 코드를 수정하고 실행하는지 어려움을 겪는다면

ConsoleApplication3에서 마우스 오른쪽 버튼 클릭 - Export ~~를 누르면 창이 뜨고, start를 눌러주면 소스코드 및 프로젝트가 해당 경로에 저장이 된다. 그 경로에서 cs파일을 눌러서 위에 한 줄 코드를 추가해주고 프로젝트를 열어서 빌드 -> 디버깅을 해주면 프로그램이 실행된다.



다음과 같이 패스워드가 출력이 되었고, 확인해보았더니



위와 같이 성공메세지가 출력되는 것을 볼 수 있다.

답은 Leteminman