

rev-basic-1

```
3  char v4[256]; // [rsp+20h] [rbp-118h] BYTE
4
5  memset(v4, 0, sizeof(v4));
6  sub_1400013E0("Input : ", argv, envp);
7  sub_140001440("%256s", v4);
8  if ( (unsigned int)sub_140001000(v4) )
9      puts("Correct");
10 else
11     puts("Wrong");
12 return 0;
13 }
```

이번에도 동일하게 조건 검사하는 코드

```

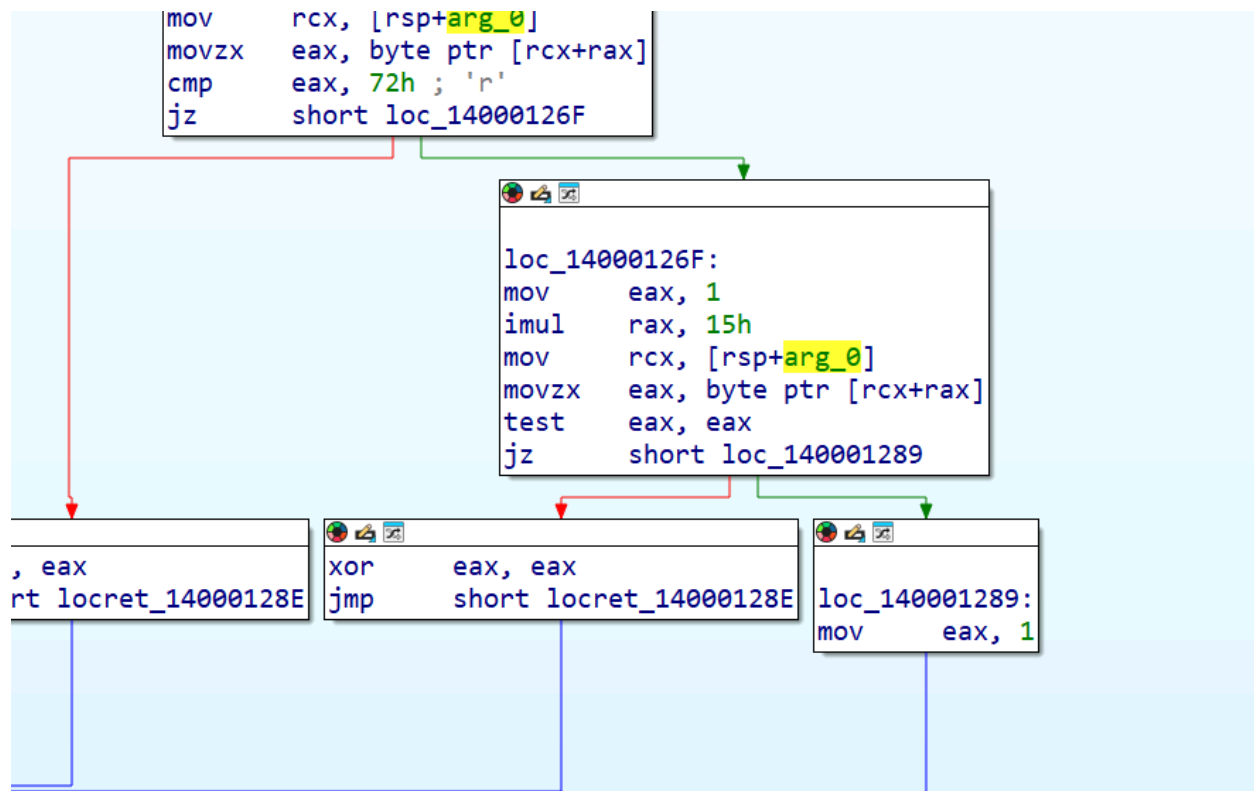
2 {
3   if ( *a1 != 67 )
4     return 0;
5   if ( a1[1] != 111 )
6     return 0;
7   if ( a1[2] != 109 )
8     return 0;
9   if ( a1[3] != 112 )
10    return 0;
11  if ( a1[4] != 97 )
12    return 0;
13  if ( a1[5] != 114 )
14    return 0;
15  if ( a1[6] != 51 )
16    return 0;
17  if ( a1[7] != 95 )
18    return 0;
19  if ( a1[8] != 116 )
20    return 0;
21  if ( a1[9] != 104 )
22    return 0;
23  if ( a1[10] != 101 )
24    return 0;
25  if ( a1[11] != 95 )
26    return 0;
27  if ( a1[12] != 99 )
28    return 0;
29  if ( a1[13] != 104 )
30    return 0;
31  if ( a1[14] != 52 )
32    return 0;
33  if ( a1[15] != 114 )
34    return 0;
35  if ( a1[16] != 97 )
36    return 0;
37  if ( a1[17] != 99 )
38    return 0;
39  if ( a1[18] != 116 )
40    return 0;
41  if ( a1[19] != 51 )
42    return 0;
43  if ( a1[20] == 114 )
44    return a1[21] == 0;

```

```
00000400 sub 140001000:1 (140001000)
```

이번에는 그래도 조건이 좀 많다.


문자열의 문자들을 검사하는 코드 같음.



모든 조건이 다 맞아야만 140001289로 점프하는 듯 보이는데... 근데 사실 이거 아이다 뷰에서도 아스키 확인 가능하다

시작부터

Compar3_the_ch4ract3r

 **Beginner rev-basic-1**
문제를 해결했습니다.

대단해요. 문제를 어떻게 해결하셨나요?

프로그래밍을 하다 보면 이런 문제를 많이 접하게 됩니다.