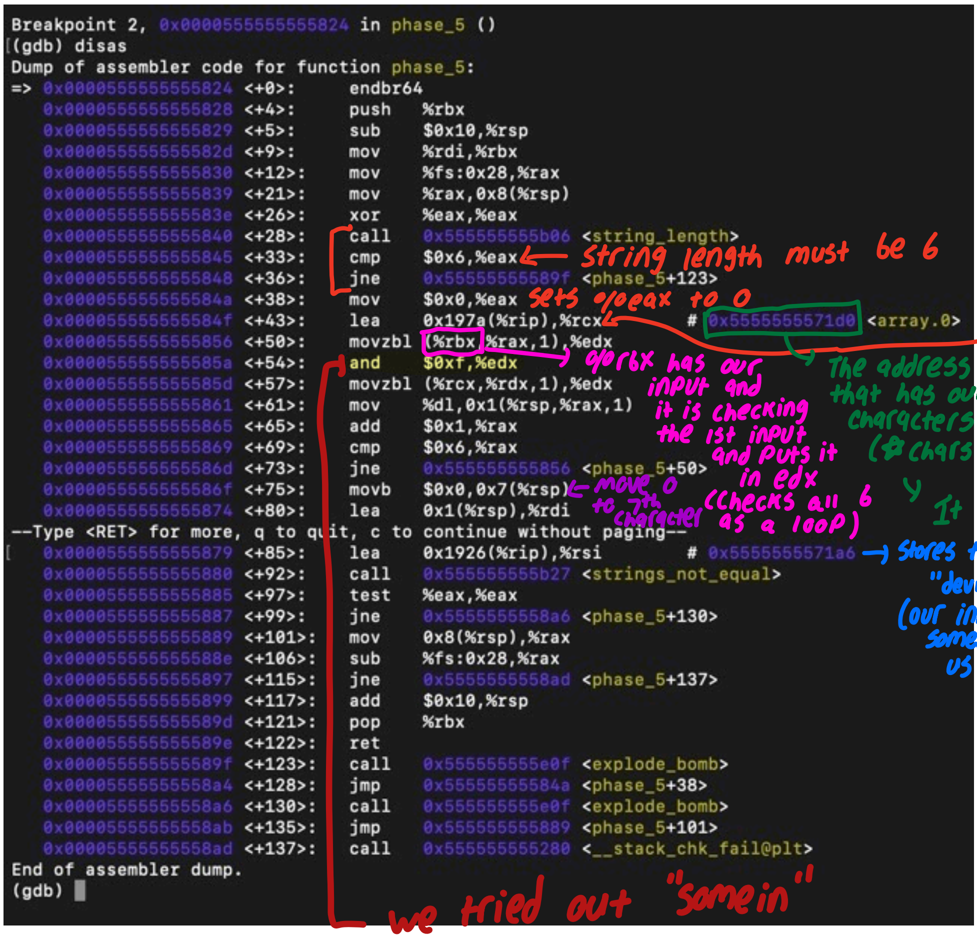

```
Breakpoint 2, 0x0000555555555824 in phase_5 ()
[(gdb) disas
Dump of assembler code for function phase_5:
=> 0x0000555555555824 <+0>:     endbr64
   0x0000555555555828 <+4>:     push   %rbx
   0x0000555555555829 <+5>:     sub    $0x10,%rsp
   0x000055555555582d <+9>:     mov    %rdi,%rbx
   0x0000555555555830 <+12>:    mov    %fs:0x28,%rax
   0x0000555555555839 <+21>:    mov    %rax,0x8(%rsp)
   0x000055555555583e <+26>:    xor    %eax,%eax
   0x0000555555555840 <+28>:    call   0x555555555b06 <string_length>
   0x0000555555555845 <+33>:    cmp    $0x6,%eax
   0x0000555555555848 <+36>:    jne    0x55555555589f <phase_5+123>
   0x000055555555584a <+38>:    mov    $0x0,%eax
   0x000055555555584f <+43>:    lea    0x197a(%rip),%rcx        # 0x5555555571d0 <array.0>
   0x0000555555555856 <+50>:    movzbl (%rbx,%rax,1),%edx
   0x000055555555585a <+54>:    and    $0xf,%edx
   0x000055555555585d <+57>:    movzbl (%rcx,%rdx,1),%edx
   0x0000555555555861 <+61>:    mov    %dl,0x1(%rsp,%rax,1)
   0x0000555555555865 <+65>:    add    $0x1,%rax
   0x0000555555555869 <+69>:    cmp    $0x6,%rax
   0x000055555555586d <+73>:    jne    0x555555555856 <phase_5+50>
   0x000055555555586f <+75>:    movb   $0x0,0x7(%rsp)
   0x0000555555555874 <+80>:    lea    0x1(%rsp),%rdi
--Type <RET> for more, q to quit, c to continue without paging--
[   0x0000555555555879 <+85>:    lea    0x1926(%rip),%rsi        # 0x5555555571a6
   0x0000555555555880 <+92>:    call   0x555555555b27 <strings_not_equal>
   0x0000555555555885 <+97>:    test   %eax,%eax
   0x0000555555555887 <+99>:    jne    0x5555555558a6 <phase_5+130>
   0x0000555555555889 <+101>:   mov    0x8(%rsp),%rax
   0x000055555555588e <+106>:   sub    %fs:0x28,%rax
   0x0000555555555897 <+115>:   jne    0x5555555558ad <phase_5+137>
   0x0000555555555899 <+117>:   add    $0x10,%rsp
   0x000055555555589d <+121>:   pop    %rbx
   0x000055555555589e <+122>:   ret
   0x000055555555589f <+123>:   call   0x555555555e0f <explode_bomb>
   0x00005555555558a4 <+128>:   jmp    0x55555555584a <phase_5+38>
   0x00005555555558a6 <+130>:   call   0x555555555e0f <explode_bomb>
   0x00005555555558ab <+135>:   jmp    0x555555555889 <phase_5+101>
   0x00005555555558ad <+137>:   call   0x555555555280 <__stack_chk_fail@plt>
End of assembler dump.
(gdb) 
```

string length must be 6

sets %eax to 0

%rcx will point to a place in memory that gives an array of characters

%rbx has our input and it is checking the 1st input and puts it in edx (checks all 6 as a loop)

The address that has our characters (# chars are 1 byte, so we use b)

↳ It contains madwiersnfotvbyl and other stuff

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

16 characters (0-15)

move 0 to 7th character

→ stores the word "devils" (our input must somehow give us this)

we tried out "somein"

↳ s came out in %rdx before this and instruction to be 0x73 or 115.

↳ After the and, %rdx turns into 3.

↳ with the o, %rdx turns into f.

↳ with the m, %rdx turns into d

devils

2 5 12 4 15 7

beldog → Answer for phase 5

somein

0x 3fd59e

x_{10} 3 15 13 5 9 14

s = 3
r = 2
q = 1
p = 0
o = 15
n = 14
m = 13
l = 12
k = 11
j = 10
i = 9
h = 8
g = 7
f = 6
e = 5
d = 4
c = 3
b = 2
a = 1