Description: I wrote a function to generate the flag, but don't worry, I bet you can't access it!

Resources given: ELF x64-bit executable named chall

Hints:

- you could reverse engineer the function, but it's not necessary
- see if you can use any debugging tools to just call the function

When executing chall no input option is given and the message “No flag for you >:(“ is printed.

```
┌──(kali㉿kali)-[~/Desktop]
└─$ ./chall
No flag for you >:(
```

```
gef➤  info functions
All defined functions:

Non-debugging symbols:
0x00000000004003c8  _init
0x0000000000400400  puts@plt
0x0000000000400410  memset@plt
0x0000000000400420  __libc_start_main@plt
0x0000000000400430  __gmon_start__@plt
0x0000000000400440  _start
0x0000000000400470  deregister_tm_clones
0x00000000004004b0  register_tm_clones
0x00000000004004f0  __do_global_dtors_aux
0x0000000000400510  frame_dummy
0x0000000000400536  c
0x0000000000400599  f
0x00000000004005ea  win
0x00000000004006b8  main
0x00000000004006d0  __libc_csu_init
0x0000000000400740  __libc_csu_fini
0x0000000000400744  _fini
gef➤
```

If we use gdb on chall and use the command info functions we can see a function called “win” at 0x00000000004005ea

To access win we can set $rip = 0x00000000004005ea so that the next instruction performed will be at 0x00000000004005ea.

We can use c to avoid needing to step through each instruction individually. Eventually printing the flag: bcactf{W0w_Y0u_m4d3_iT_b810c453a9ac9} and then proceeding to crash.

```
                                                                    threads
[#0] Id 1, Name: "chall", stopped 0x4006bc in main (), reason: BREAKPOINT
                                                                     trace
[#0] 0x4006bc → main()

gef➤  set $rip = 0x00000000004005ea
gef➤  c
Continuing.
bcactf{W0w_Y0u_m4d3_iT_b810c453a9ac9}

Program received signal SIGSEGV, Segmentation fault.
```

Document created by: ohtheirony