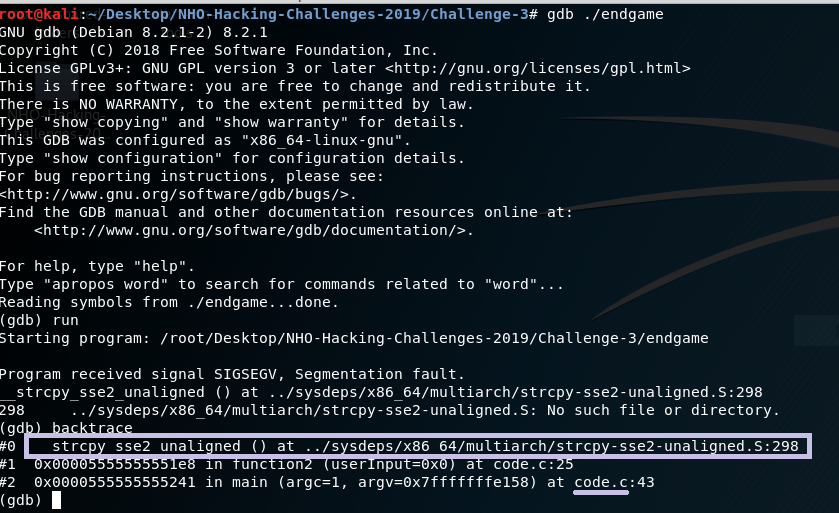
**Solution for Challenge-3**

**Observations:**

1. When the file was opened on a text editor, few blocks along with the letters “ELF” were seen.
2. Then a commands *readelf* endgame and *file* endgame were run to understand that it is PIE Executable.

**Debugging:**

1. The file was given the rights to be executed by using chmod as *chmod* +x endgame.
2. But when the file was being run, it was throwing Segmentation Fault error.
3. Then to debug the execution *gdb* was used as follows

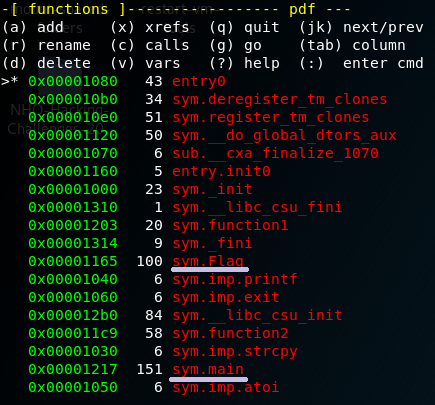


1. With this it was completely evident that it’s a C program compiled using a command like

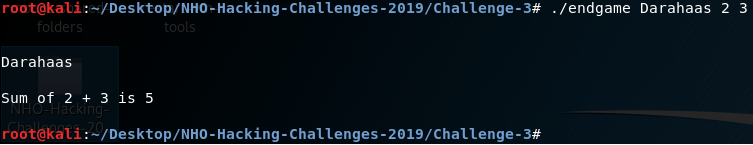
*gcc* -fPIE c\_program

and there was an error with the *strcpy* function.

1. Then upon browsing regarding how to get visibility into the program logic, I landed up on a utility called **radare2.**
2. By loading the endgame file and deassembing the binary did I find the various functions being used with the C program which are as follows.



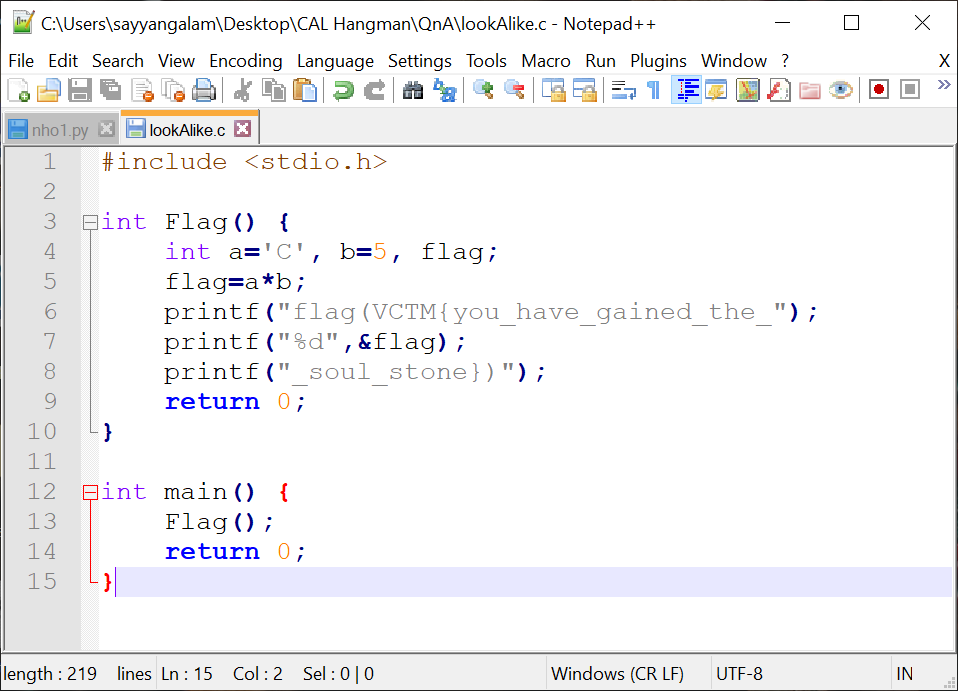
1. By having a look at the assembly code of the main function, it intuitively felt that it might be taking three parameters. After trying few inputs, the following was the result which made it evident that the main function’s logic doesn’t have anything to do with the flag.

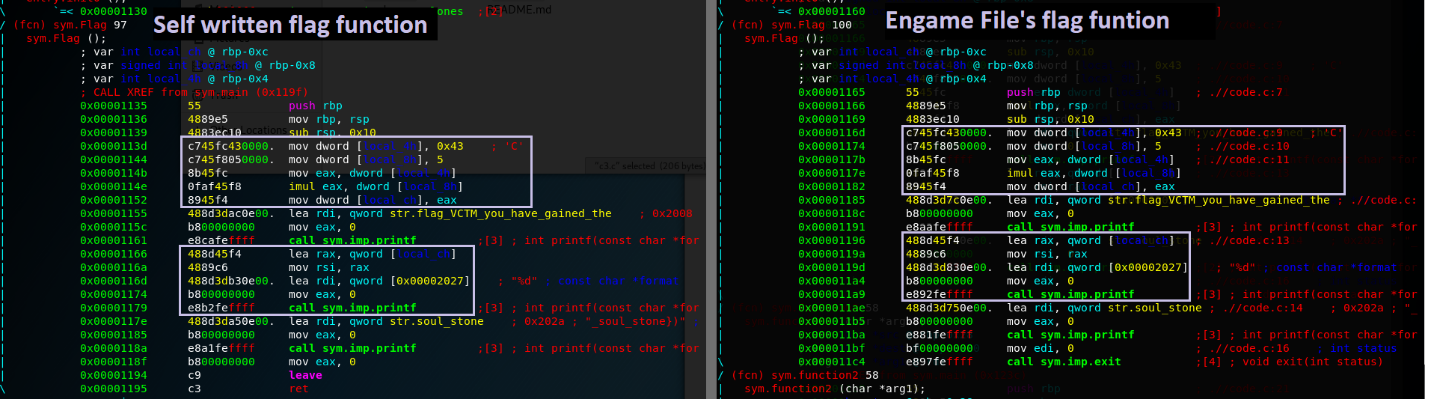


1. But along with the main function, there are other functions too in the code which includes a **Flag** function whose assembly language code looks as the follows.



1. After failing for a number of times trying to edit the assembly code, I tried writing a couple of C programs with a flag function, having a similar logic which I have observed in the endgame and I have landed up with the following results.





1. Looking at the similarities with regards to the %d part in both of them, I landed up with the following conclusion.

**Conclusion:**

The flag is flag(VCTM{you\_have\_gained\_the\_***NUMBER***\_soul\_stone})

Where the ***NUMBER*** isthe **address of the variable** which stores the number **335 (0x43\*5)** which keeps changing every time we execute the program.

**Another observation:** In python when I have applied the **chr** function on 335, i.e. chr(335), it says that the character mapped to 335 number is **ŏ**