

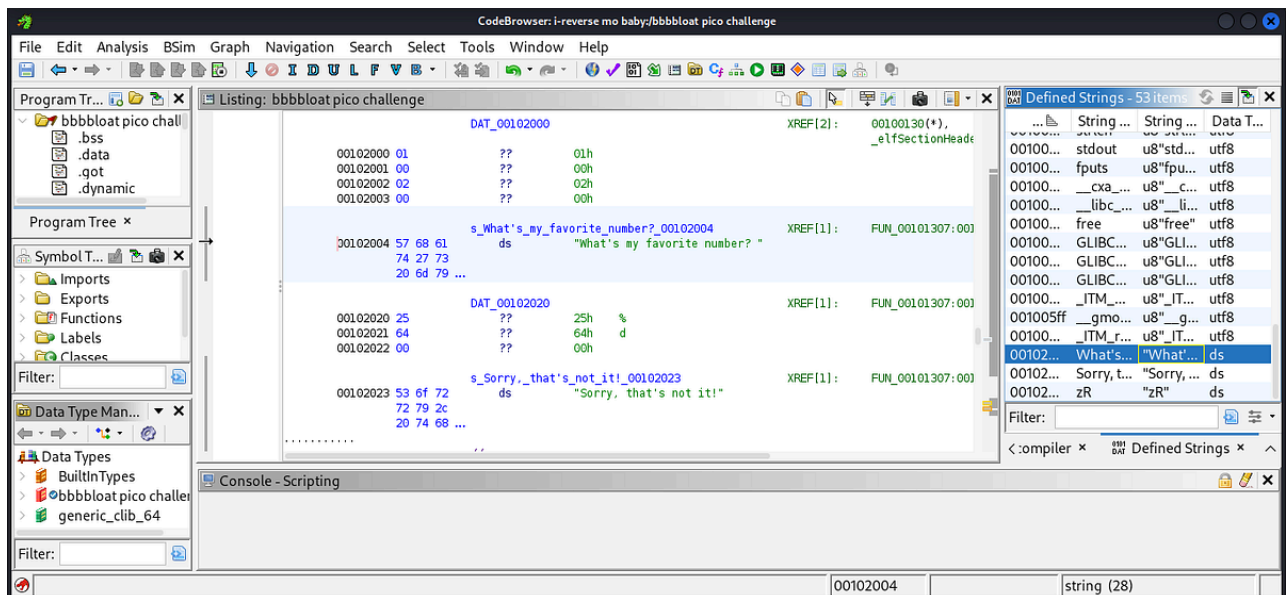
# Pico-CTF : bbbbloat challenge Solution

**Category: Reverse Engineering**

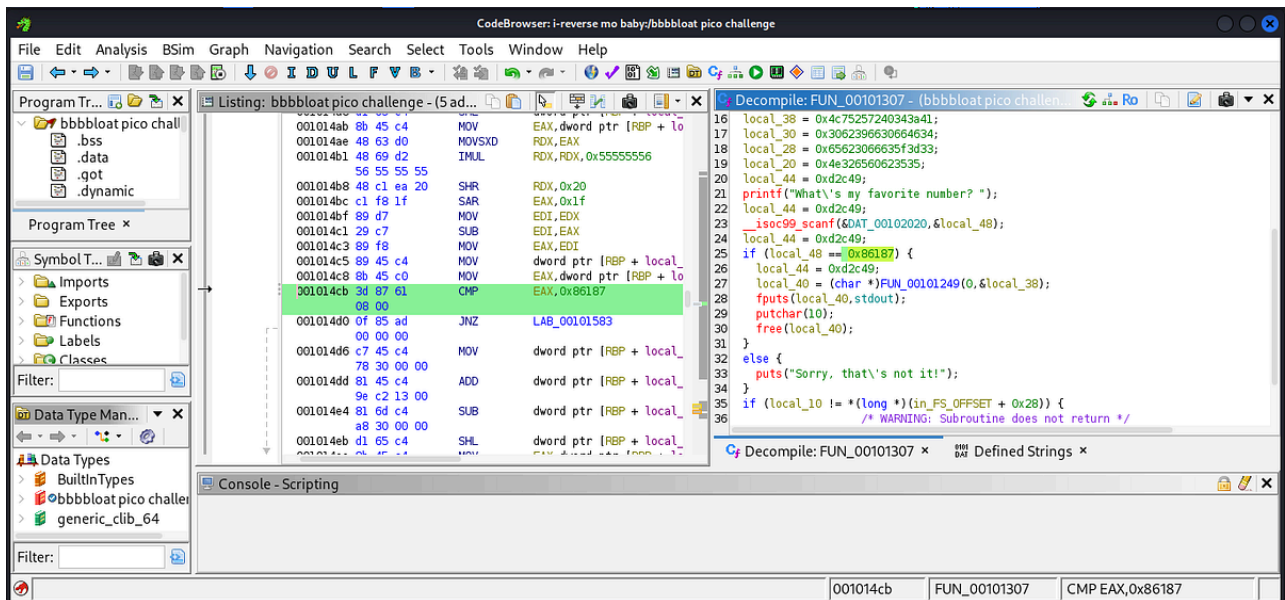
**Tool Used: Ghidra**

I started by trying to understand how the program worked. Upon inspection, I noticed that the program contains this one **function**, and within it, a key piece of code stood out.

```
if (local_48 == 0x86187) {
    local_44 = 0xd2c49;
    local_40 = (char *)FUN_00101249(0, &local_38);
    fputs(local_40, stdout);
    putchar(10);
    free(local_40);
} else {
    puts("Sorry, that's not it!");
}
```



Here's what's happening: the program checks if local\_48 matches the hexadecimal value 0x86187. If it does, it calls a function that unscrambles the flag and prints it. Otherwise, it simply outputs: "Sorry, that's not it!".

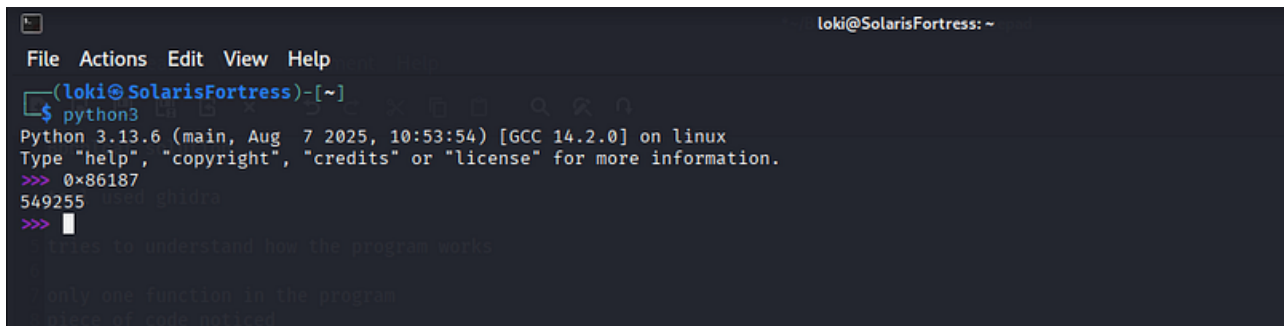


To get the correct input, I converted the hexadecimal value to decimal.

```

>>> hex_value = 0x86187
>>> int(hex_value)
549255

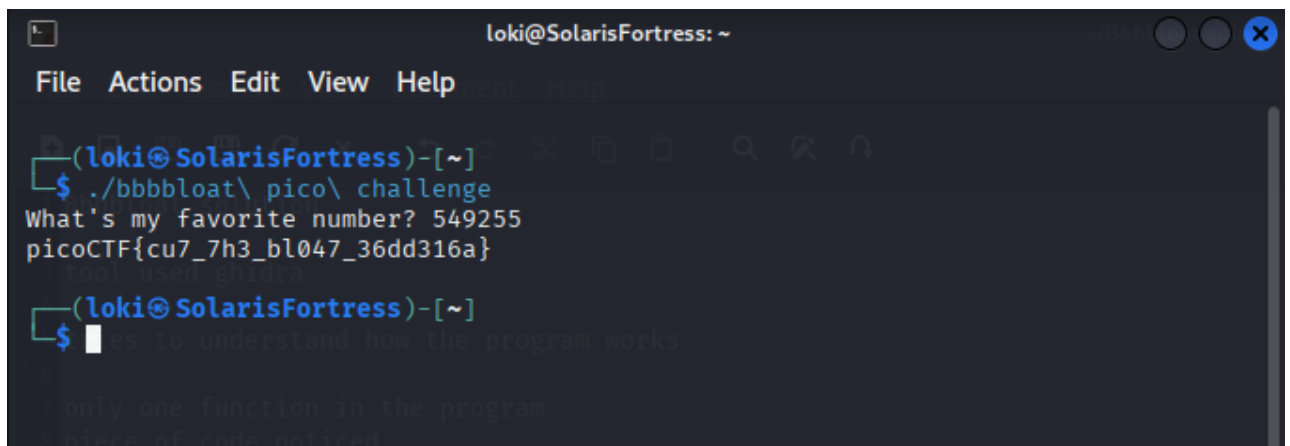
```



So, the answer to the prompt "What's my favorite number?" is 549255.

Running the program with this input reveals the flag:

**Flag:** picoCTF{cu7\_7h3\_blo47\_36dd316a}



The screenshot shows a terminal window titled "loki@SolarisFortress: ~". The window has a menu bar with "File", "Actions", "Edit", "View", and "Help". The terminal content shows a user running the command `./bbbbloat\ pico\ challenge`. The program outputs "What's my favorite number? 549255" and "picoCTF{cu7\_7h3\_bl047\_36dd316a}". The user then enters a prompt `(loki@SolarisFortress)-[~]` followed by a dollar sign `$` and a cursor. Faint, semi-transparent text in the background of the terminal reads: "tool used ghidra", "es to understand how the program works", "only one function in the program", and "piece of code noticed".

By [Alexander Sapo](#) on [November 15, 2025](#).

[Canonical link](#)

Exported from [Medium](#) on February 7, 2026.