A **heap overflow** happens when a program writes more data to the heap (dynamically allocated memory) than it was allocated. This can lead to **data corruption**, **program crashes**, or even **remote code execution** if exploited maliciously.

The goal is to overflow the heap memory

Welcome to heap0!

I put my data on the heap so it should be safe from any tampering.

Since my data isn't on the stack I'll even let you write whatever info you want to the heap, I already took care of using malloc for you.

Heap State:

+-------------+----------------+

[*] Address -> Heap Data

+-------------+----------------+

[*] 0x55d640d072b0 -> pico

+-------------+----------------+

[*] 0x55d640d072d0 -> bico

+-------------+----------------+

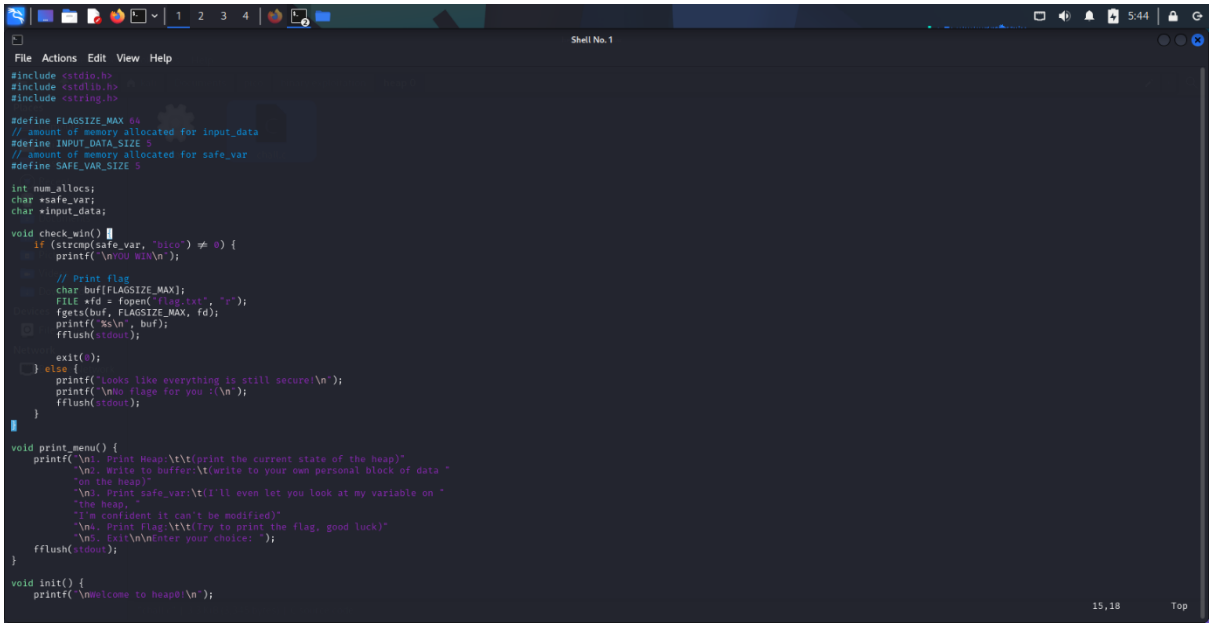
1. Print Heap: (print the current state of the heap)

2. Write to buffer: (write to your own personal block of data on the heap)

3. Print safe_var: (I'll even let you look at my variable on the heap, I'm confident it can't be modified)

4. Print Flag: (Try to print the flag, good luck)

5. Exit



If the safevar variable is changed then we will get the flag so use option 2 write to buffer and input a long input

Enter your choice: 2

Data for buffer: ABCDEFGIJKLMNOPQRSTUVWXYZ1234567890

```
Heap State:

+-------------+----------------+

[*] Address   ->   Heap Data

+-------------+----------------+

[*]   0x55d640d072b0  ->   ABCDEFGIJKLMNOPQRSTUVWXYZ1234567890

+-------------+----------------+

[*]   0x55d640d072d0  ->   890


Enter your choice: 4


YOU WIN

picoCTF{my_first_heap_overflow_4fa6dd49}
```