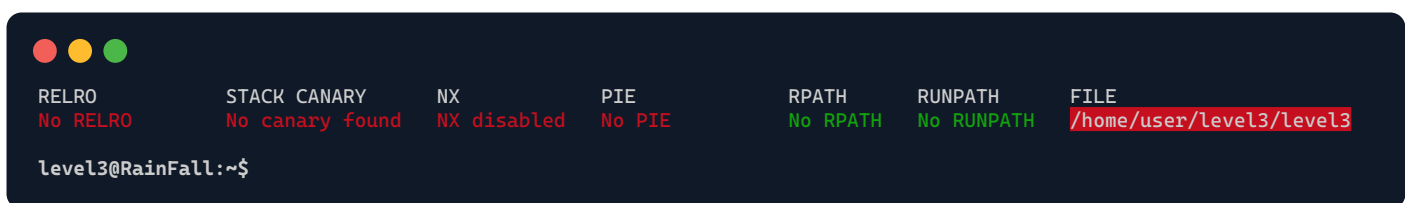


# ./level3



Decompiled file with **Ghidra**:



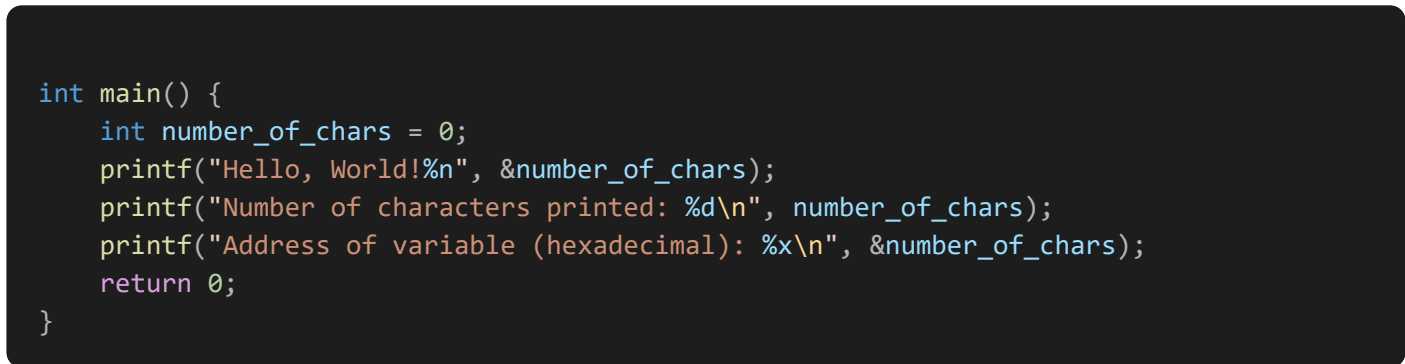
In the function **v(void)**, the program captures our input into the buffer and then echoes it using **printf**. If we manage to set the global variable **m** to 64, the program hands us shell access.

## Attack Vector:

**Format String Exploitation:** we can use the **printf(buffer)** vulnerability to inject *format specifiers* into our input. In particular:

1. **%x**: This specifier lets us read and display memory content. By chaining these, we can read sequential memory addresses on the stack.
2. **%n**: While most specifiers read data from memory, **%n** uniquely writes to it. This specifier captures the number of characters printed so far and stores it in the located at the memory address provided as an argument. This can be exploited to overwrite particular memory locations with chosen values.

A practical example of how **%n** and **%x** work:



However, when **%x** is used in a format string without a matching argument, it reads and displays the next value on the stack.

So, we can write the address of the **m** variable into the input buffer. Then, by using the **%x** specifier we cycle through the stack's contents and approach the injected address of **m**.

Once at the correct location, we'll use the **%n** specifier to overwrite its value. To set the value to 64, we must factor in the characters already produced by the **%x** specifiers. The remainder can be filled with **As**.

