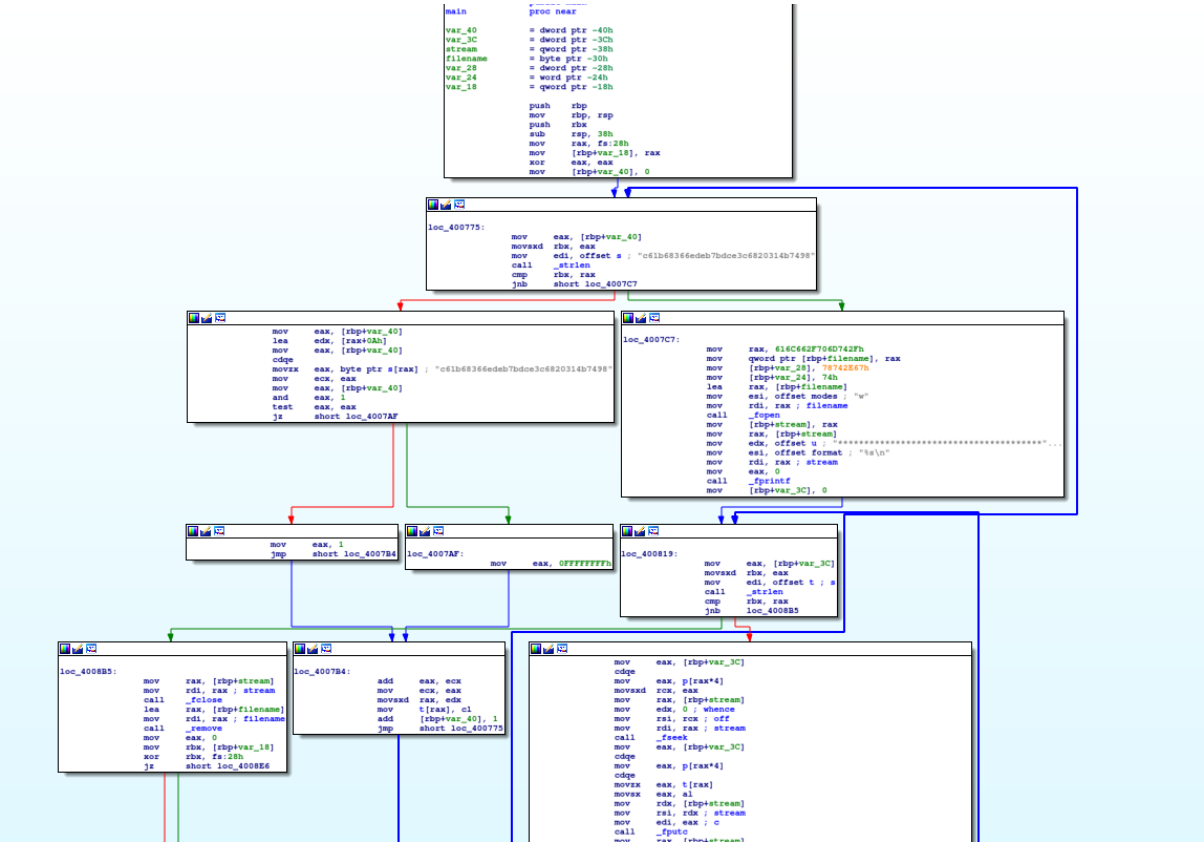
5_getit

Running the program, nothing happens. We can then open the binary with ida and inspect it.



We can see that the program starts with a strlen, and compare it with rbx, initially 0. After some operations, there is and add [rbp+var_40], 1 and it is moved to rbx. This is a cycle, that goes from 0 to the len of the string s. When the cycle is over, there is a second cycle working with files, and then the program ends. Let's start analyzing the first cycle. We can see that we have a string s, and at each iteration the index of iteration contained at rbp+var_40 is added to 0Ah and then moved in eax. This value is used as an index of the string s. The char is picked, transformed with some operation, and stored to another string t with the instruction t[rax], cl. We don't really need to understand what the cycle is doing, we can just put a breakpoint at this instruction to see how the string t is populated. Looking at it in the memory, we can see that the string is in the format SharifCTF{??????????????????????}, and at each iteration a new char replaces a '?'. So, if we put a break point at that instruction and loop 32 times, we will see the flag populating at each iteration, obtaining the final one: SharifCTF{b70c59275fcfa8aebf2d5911223c6589}

Alternatively, we could have analyzed the second cycle, which basically is taking in input the string t, and writing it on a file. It would have been enough to put a break point at the beginning of this second cycle, since the flag was already created. Putting a breakpoint at the strlen at address 0x400824, we can see at rdi the offset of the string t (0x6010e0), and so the flag by inspecting it. To inspect an address, if double click on the name/address

doesn’t work, we can use the shortcut g to go to an offset, and insert the address to see the memory.