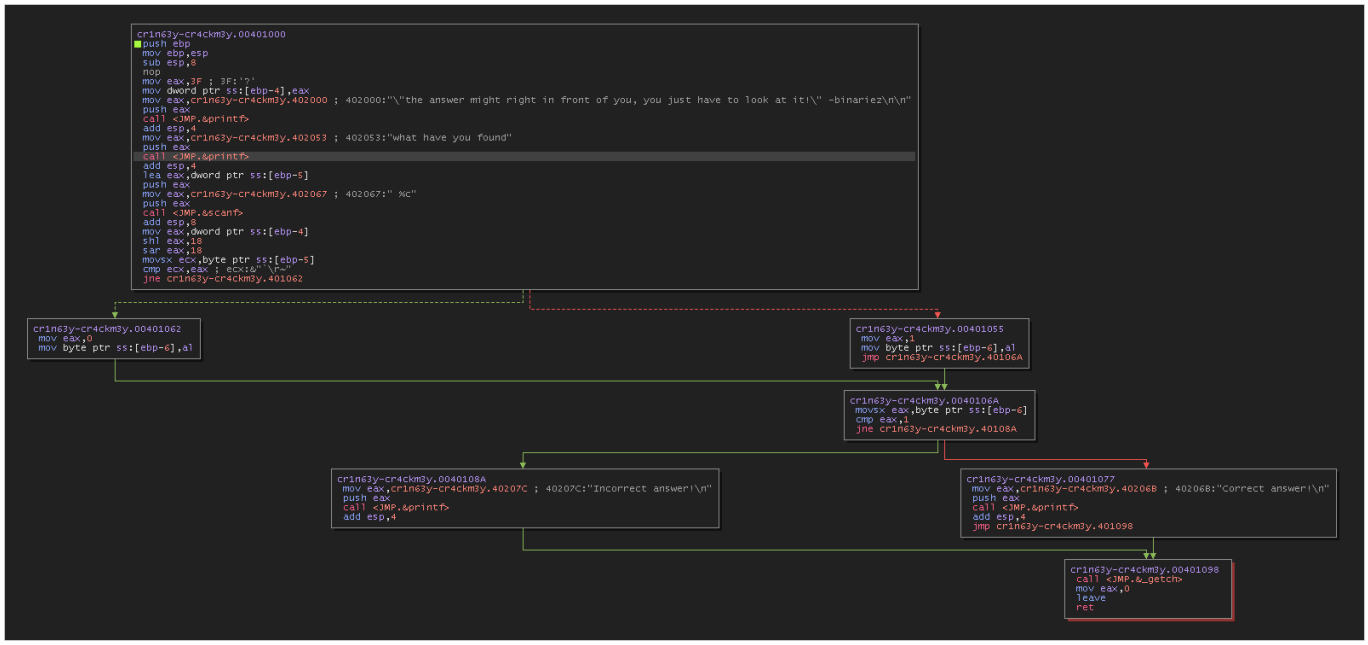
Cringey Crackme

This one was pretty simple to solve, and it was indeed pretty cringe...

Looking at the binary we can see our main function starting at `0x401000` (base image 0x400000), the binary was compiled for x86 systems and it was compiled using the `Tiny C Compiler` which is a "tiny" compiler to run programs on slower systems.



We can see the value `0x3f` being moved inside `eax` and then into `ss:[ebp-4]`, looking a little deeper we can see that this value is being compared with our input `ss:[ebp-5]`, just before it we can see 2 instructions `shl and sar`, these instructions are basically useless since we shift it left and we shift it right which cancels the left shift, you can confirm this by writing the following program and then debug it.

```
.global _start
.intel_syntax noprefix
_start:
        mov eax, 0x3f
        shl eax, 0x18
        sar eax, 0x18
```

```
# compiled with : gcc -w -nostdlib -static -m32 file.s -o file
```

So we do the most logical thing which is to try inputting the character `0x3f (?)` to our program, and we win the challenge.