

./level03



Decompiled file with *Ghidra*:



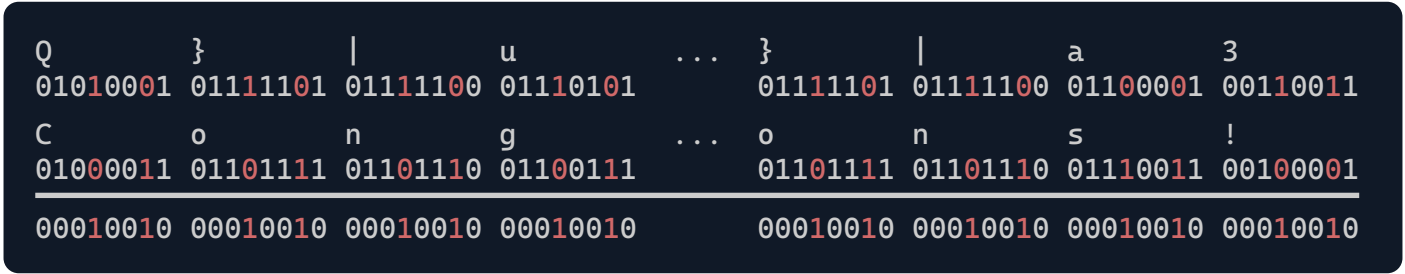
This C program is a simple password checker that uses a cryptographic **XOR operation** for validation. It begins by asking for an integer password from the user. Internally, it takes the user input and calculates the difference from the hexadecimal constant 0x1337d00d. This difference is then used as a **key** to decrypt a hardcoded cipher text.

The valid range for the **key** is limited, as indicated by the conditional checks in the program: it must be between 1 and 21, inclusive. If the difference doesn't fall within these ranges, the program will use a random value as the key, which typically results in decryption failure and an Invalid Password! message.

The decryption process involves a bitwise XOR operation (exclusive OR), a simple bitwise operation that gives 0 if the bits are the same, and it gives 1 if the bits are different.

The encrypted string in the program is **Q}|u`sfg~sf{}}|a3**. If, after being XORed with the **key**, it matches **Congratulations!**, the program opens a system **shell**.

To crack the program, we need to *reverse-engineer* the **key** from the known plaintext and the encrypted string. By XORing these two strings, we obtain the key:



The key is 10010_2 (12_{16}) and can then be used to find the correct password: it's the number that, when subtracted from 0x1337d00d, yields the key.

