Main challenge

1. 00401190: Start the program and print some information. Then calls 004010F9
2. 004010F9: Gets the username and serial and checks that they are not Null. If the username was Null then it goes to 00401151. If the serial was Null it goes to 00401160. If everything is ok it calls 004010C0. Then checks whether the return of the call is 0x3h or not and if so calls 00401049 to do the calculation for the username and 00401000 to check it with the serial that the user inputted.
3. 004010C0: Checks if the serial is 12-string long. If the call returns 0x3h it means that everything was ok.
4. 00401049: in the first rows of this call calculate the length of the username entered by the user and it stores it in the ECX flag. In the first loop, it calculates $\Sigma((Hex-of-the-username-char)*\ (place-of-the-char))$ and stores this number in the EBX flag. Then calculates the following $(EBX\ \odot\ 13131313\ \otimes\ 1234ABCD)$ and stores odd places of this number in EBX and even places in EAX. Then SHR of character of EBX. Stores these two numbers in 4020A0 and 4020A4. In the next loop, it checks if the most upright character of EAX is above 0x9h and if so it ORs it with 0x30h. If not, it adds 0x37h to it.
5. 00401000: It loads the calculated EAX and EBX and compares them with the serial input of the user. The format of the serial is **-****-**** and the program checks the red parts with EAX and EBX.