Challenge 1:

1. 00401000: It serves as the main function. After calling 0040110E the program saves module entry points at 00403008, 0040000 (which is the start address of the PE header) at 0040300C, 400 at 00403010, and 600 at 00403014. After that, the program calls 00403019. Then the program asks for the user input which is the serial. Then the program calculates the input length of the user input and checks whether it is 8 characters long or not. If so then it loads the first four characters of the user input in the EAX and compares it with 504E4D41 which is AMNP (in ASCII format). Then it loads the second four characters of the serial in EBX and loads another hex number from its memory to ECX. Then XOR's the first character of the EBX with the third character ECX, the second one with the third one, the third one with the first one, and lastly the fourth one with the second one. The ECX was 90238951 respectively. Then the program XOR's all the calculated numbers with 0x0Dh (in the 00403019 part we describe how this number is calculated) and compares it to 6DB278DB. Because the bit-by-bit XOR is a reversible action we can easily calculate that the intended serial was 16FV. So the intended serial was AMNP16FV.
2. 0040110E: It checks some of the PE headers and if they are wrong it breaks the call. If everything was right it creates a heap. If the heap is created then the EAX is the handle of that heap and if the EAX is NULL after that it means that the heap was not created. The program then

stores EAX(handle of the heap) in its memory located at 00403004. If anything goes wrong it makes the EAX zero which in turn makes an error and if everything goes right it makes the EAX one and the program continues.

3. 00403019: in this part, the program takes a snapshot of the running executables in the system and looks for these executables.
    - ollydbg.exe
    - ollydbg 9 in1.exe
    - ust_2bg.exe
    - derox.exe
    - ollyice.exe
    - idaq.exe
    - idag.exe
    - windbg.exe
    - immunitydebugger.exe.

   If the program finds that any of these programs are running in the background, it will add 0x5h to the hex number in the location of 00403018(which at the start of the program is equal to 0x0Dh). For patching of the program, we can change the 004011F5 to NOP or we can add the number 0 instead of 0x5h to 00403018. By doing so there is no difference if these programs were running or not. The serial will always be the same.