**Task 2**

**Group: Isaias Leos, David Amparan, and Alex Vasquez**

Complete the DLLInject Template

Your task write in the correct values in the DLL\_Inject source code to toggle the unlimited ammo cheat for doom.

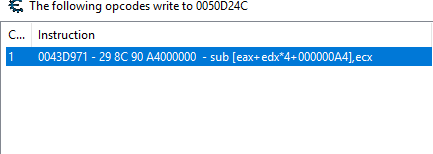
1. Our first task is to find the memory address that controls the ammunition of the gun. If we modify the reduction caused by firing the weapon then we will not run out of ammo



* 1. After attaching Cheat Engine onto chocolate doom, we being firing our weapon. Originally, we began with 45 rounds (we shot off some while testing for the address).

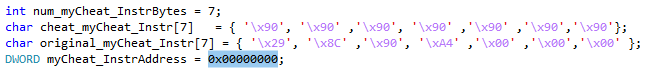


* 1. We enter 45 in the scanner, within Cheat Engine, to find all addresses with the value 45. We then fire the gun again and pin point the addresses that update this change which is now 44.
  2. We then find 3 possible addresses for ammo consumption. Let us choose 0050D24C and see what writes that memory to possibly cause a change of ammo. Firing again we see that instruction 0043D971 writes into this address with a subtraction command



* 1. Finding the address and the instructions to modify it allow us to write the unlimited ammo cheat since we know the location and the original code which are needed for the template.

1. Our second task is to write the acquired information into the Dll\_Inject template
   1. Our first task is to convert the code that writes into the memory into shellcode. We can do this with an instruction disassembler/assembler online. The code that writes into the ammo memory is: sub [eax+edx\*4+000000A4],ecx :
   2. This translates into: '\x29', '\x8C' ,'\x90', '\xA4' ,'\x00' ,'\x00','\x00' which is put into original my cheat instruction filed on CommandWindow.h file. within cheat\_myCheat\_Instr, which is our cheat, we implement seven \x090. These are NOP instructions which means that our ammo will not be changed.
   3. num\_myCheat\_InstrBytes, cheat\_myCheat\_Instr, and original\_myCheat\_Instr are set to sizes of 7



* 1. Finally, for myCheat\_InstrucAddress we will put 0043D971 since that is the instruction writing into the ammo memory



