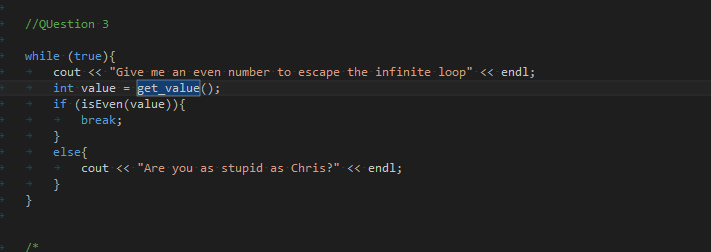
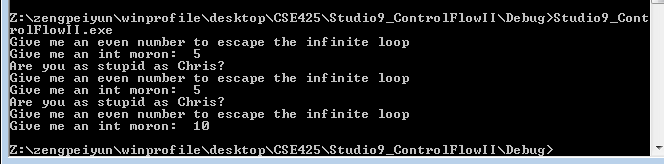
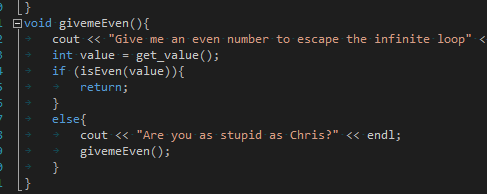
1. Amazing Seed; Creepy Chris; Whatever Mengna
2. What we notice is that everything is run and evaluated sequentially. That means that our program proceeds line by line, thus if there is a cout statement followed by another cout statement, we can expect the first cout statement to be printed out first and then the second one.



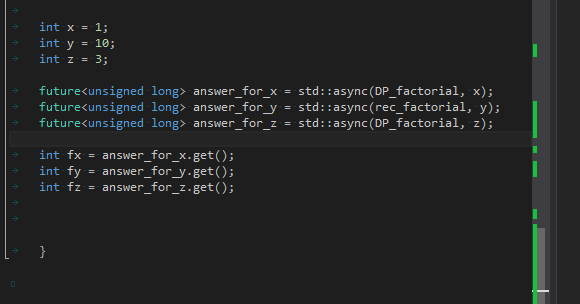


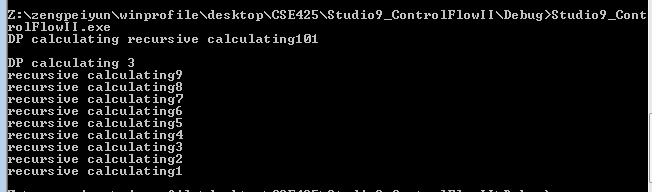
4.



5. It is near impossible to tell which thread gets launched first since we are not waiting for the threads at the end of main we can not rely on print out to accurately print what threads have been launched. Indeed when we tried to print out from within our function we would get 0 or 1 character only to print before the main thread shutdown and therefore closed the stream. This rendered us unable to determine what threads had been launched and in what order.

6.





We do see interesting behavior, we get a print out that appears to be jumbled. Given the context that this program is multithreaded and thus running from different threads when a thread tries to write out using std::cout, if another thread too tries to write out using std::cout the two will overwrite or conflict with each other. Thus this is the jumbled mess that we see above.