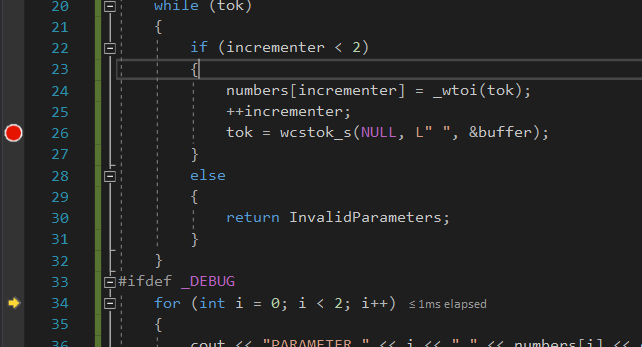
Evidence of Debugging

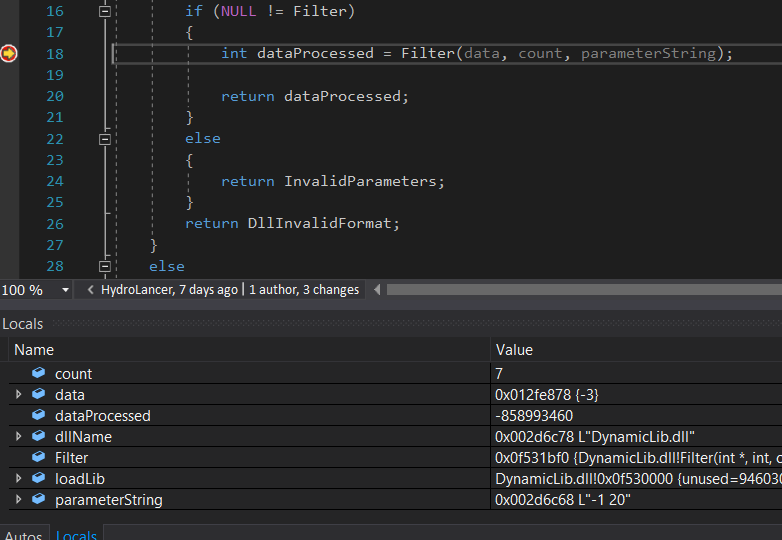
# Breakpoints

1.



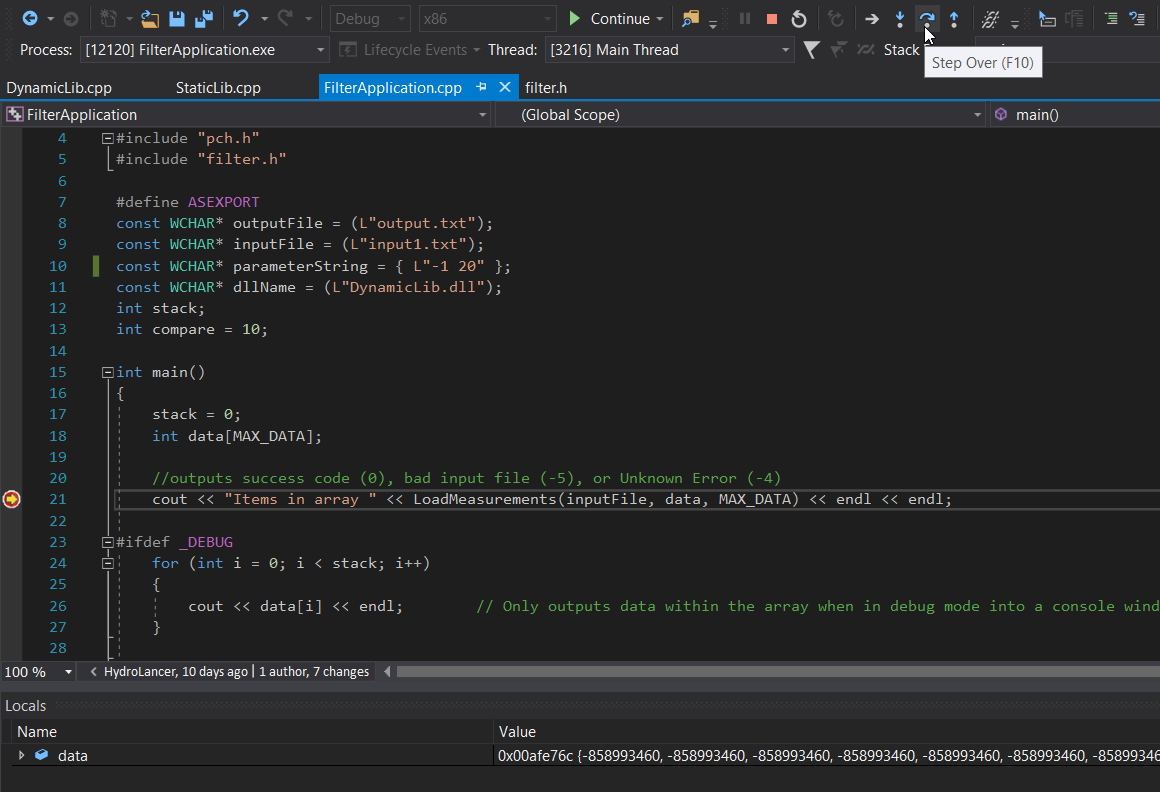
This breakpoint was placed into the DLL, as I was having issues understanding why my tokenizing code wasn’t working 100%. Combined with a variable watching, I could then understand using step by step debugging where it was going wrong.

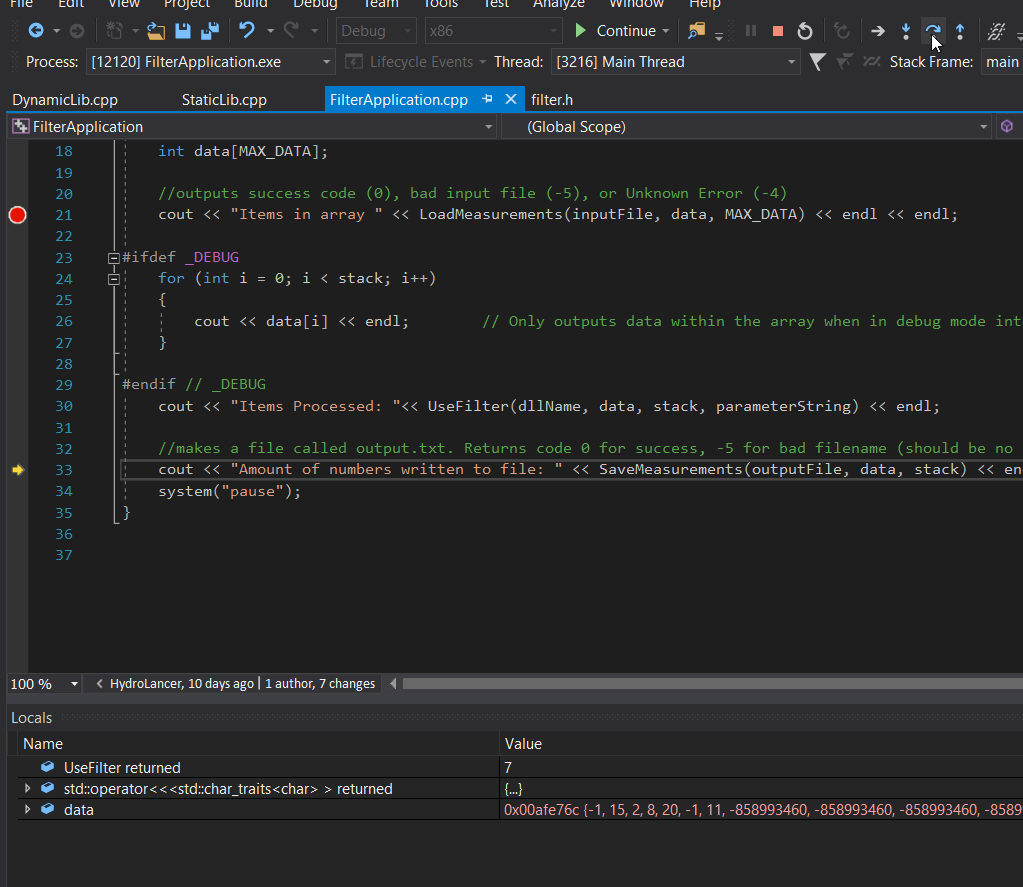
2.



2 was used to double check that all the variables and the function pointer were being assigned as intended when loading the DLL into memory.

# Step Over





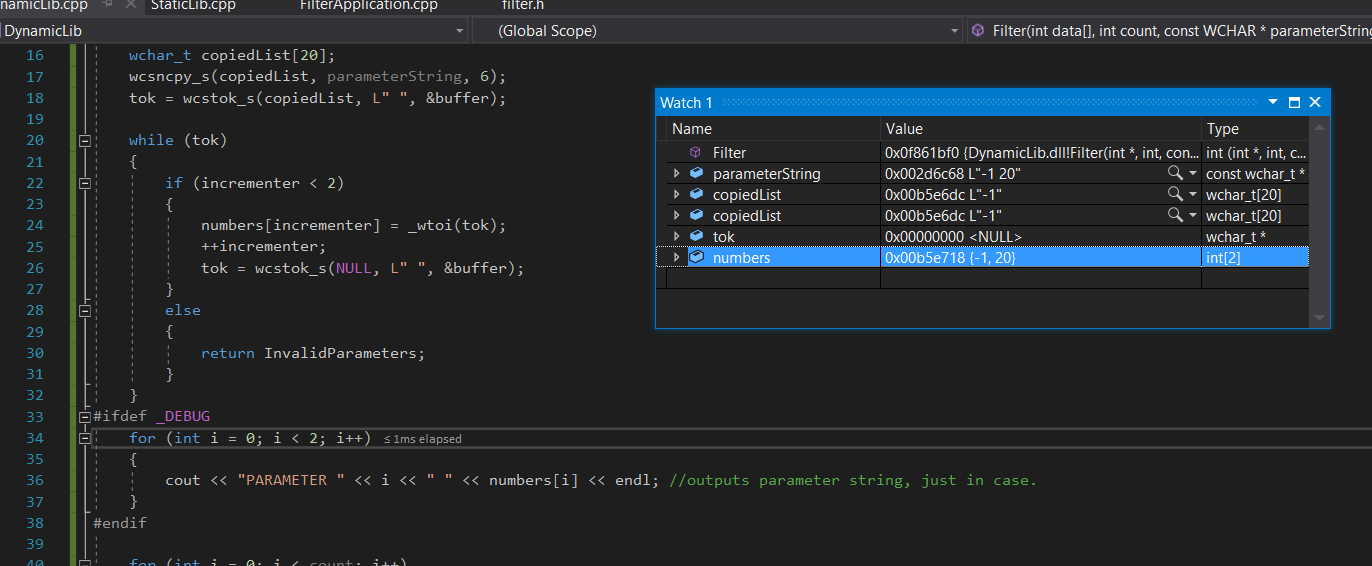
As demonstrated in the above 2 pictures, step over was used to “skip” over parts of code, to then be able to step into another part of the file, while having a breakpoint much earlier in the code so that variables could be watched as I stepped through the program. Step over simply executes the code within a function call, without actually going into it to step through each line manually.

# Variable Viewing (Watch) & Step Into

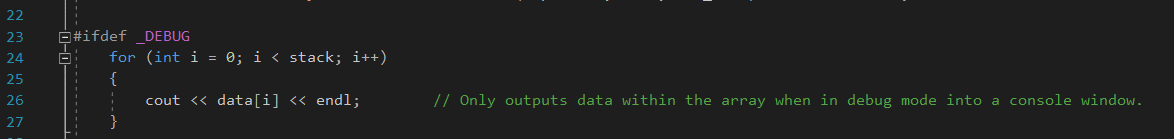
# 

This is a watch window on the variables above. – I placed a watch on these, just to make sure that the tokenizing part of the Filter Algorithm was correctly taking apart the numbers given to it in the WCHAR\* constant ‘*parameterString’* and successfully placing them into the ‘*copiedList’* integer array.

This also includes **Step into** to ensure that the rest of the parameter string was filled in correctly, and it didn’t skip over to the next function until it was ready. Step into simply moves into the next line of code during break, and doesn’t skip over any function calls.

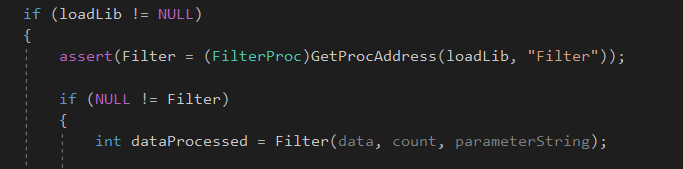


# Preprocessor to support debugging without dynamic debugger

This particular *#ifdef \_DEBUG* was used to display the contents to the input file after it had been placed into the array without having to use a breakpoint to view the variables by a watch window.

***#IFDEF \_DEBUG*** is a section of code, usually when displaying variables of some kind, which will only display when built in debug mode. When the compiler is in release mode, any code between #ifdef \_Debug and #endif would be ignored.

# Example of Assert use



Using *assert(condition)* in this project allowed me to 100% make sure the program wouldn’t execute any further unless the function pointer is correctly assigned.