This lab assignment will introduce a debugger called GDB. You will learn how to step into functions, and you will get an even better understanding of what gets passed into functions when the parameters are call by value vs. call by reference.

1. Create a subdirectory called **`lab7`**
2. Copy the program min_sum.cc from Blackboard.
3. Navigate to your lab7 folder in a terminal (this can be in VS Code if you want to see the whole source file as you are debugging).

4. Debugging with GDB

a. You will be stepping through `min_sum.cc` line by line using the debugger
b. Compile your program using the –g option:
   g++ -g –Wall min_sum.cc

   The "-g" adds some additional information into the executable that helps the debugger give you more useful output.
c. Run your program in gdb:
   gdb ./a.out
d. Set a breakpoint on line 18
   i. Type:
      break 18
e. To begin executing the program use the command:
   run
f. The program should execute up to line 18 which should be showing on the screen indicating to you that this line will be executed next.
g. *Step over* line 18 and line 20 using the "next" or "n" command. This will execute those two lines.
h. **Answer question 1 on the answer sheet**
i. *Step into* the function `get_num()` by using the "step" or "s" command.
   If you use the "n" command here, you will not be able to see what is happening in `get_num()` as it executes.
j. Examine the address of the variable num by using the command "p num"
k. **Answer questions 2 and 3 on the Answer Sheet**
l. *Step over* (lines 40 and 41) and enter 15 when you are prompted for a number.
m. *Step over* line 42 to go back to the main program (line 24 should be showing to be executed next but do not execute this line yet)
n. **Answer question 4 on the answer sheet**
o. *Step over* twice
p. *Step into* `find_min` and `sum_all` functions the same way as before
q. Answer questions 5-7 on the Answer Sheet

Look at the local variables, watch them change as you move through the code when you choose to *Step over*. You may find the following two commands useful:

"info locals" - show names and values of all local variables currently in scope
"info args" - show names and values of all function arguments currently in scope.

If you wish you can continue the program execution (without stepping through each line of code) by using the "continue" or "c" command.

***Continue* will run the code and stop at the next breakpoint or run the rest of the code if there are no additional breakpoints.**

Finish by typing -1 to end the program.

r. When the program ends and you are finished debugging, use the "quit" or "q" command to exit the debugger.

5. (30 pts) **Rewrite** the Single value returning function `sum_all` as a **`void`** function. The declaration (prototype) will be as follows:

```
void sum_all(int &cur_sum, int cur_num);
```

Don't forget to change the **function call in the main function**. Compile and run the program with several values to make sure your changes are correct.

6. Submission:

Turn in your answer sheet and edited code on Blackboard.