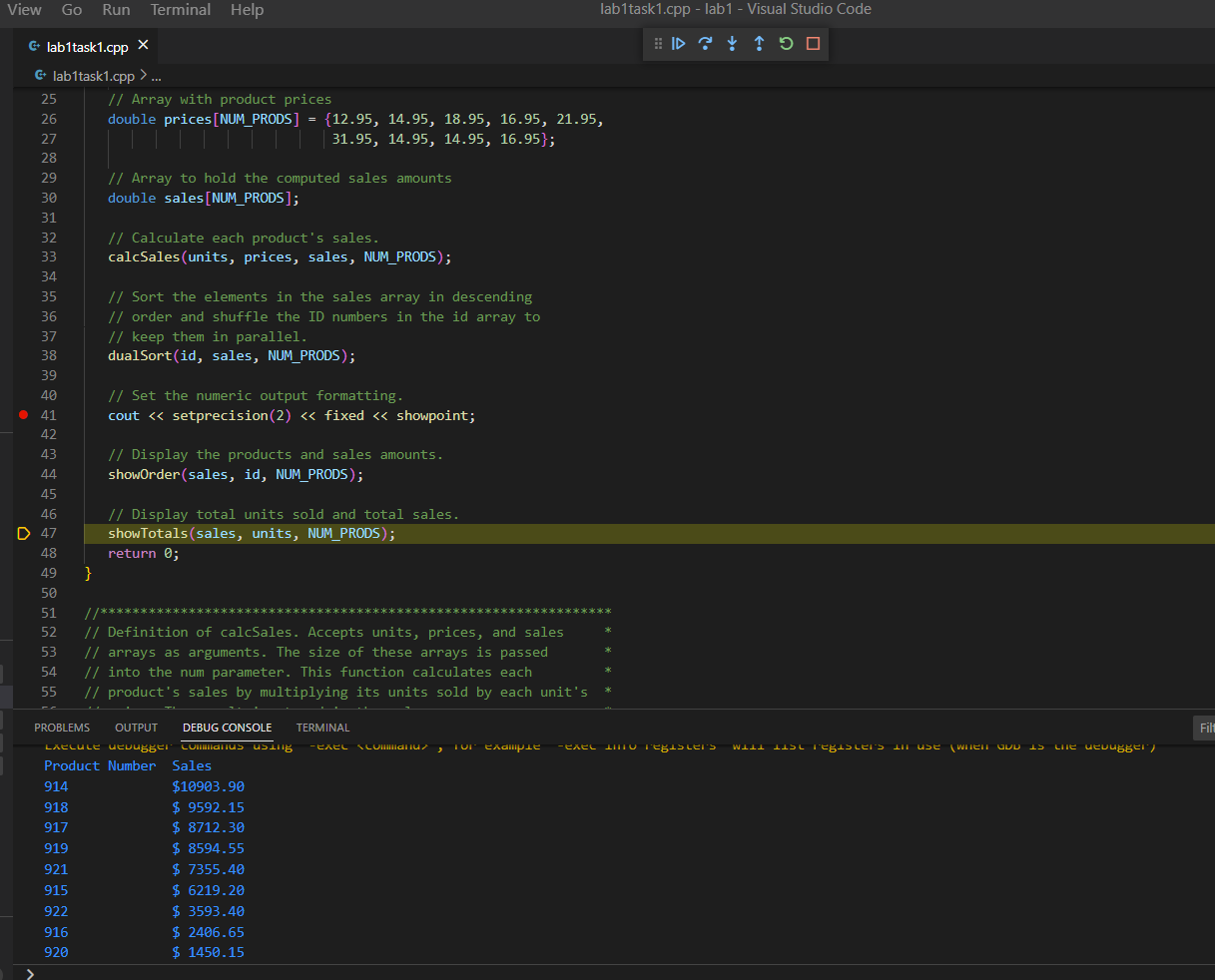
Fiona O’Connell

Ayushi Sharma

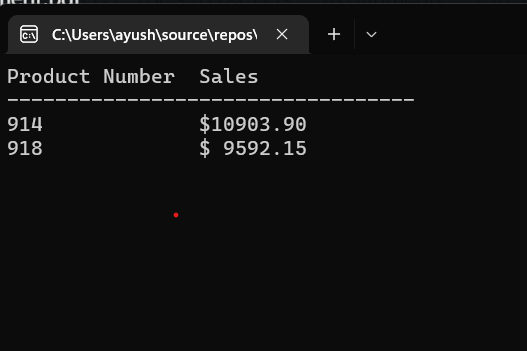
Data Structures Lab 1

Task 1:

In this assignment we were able to familiarize ourselves with the Visual Studio Code environment, as well design and implement structures in C++. We practiced debugging, utilizing tools such as breakpoints, and identifying possible problems in the code in order to fix them. Being able to read and understand how a program works is extremely important for a career in CS. Bugs will always show up when writing/developing code and testing programs so we need to know how to effectively manage them.



(Task 1, set breakpoint and step through code)

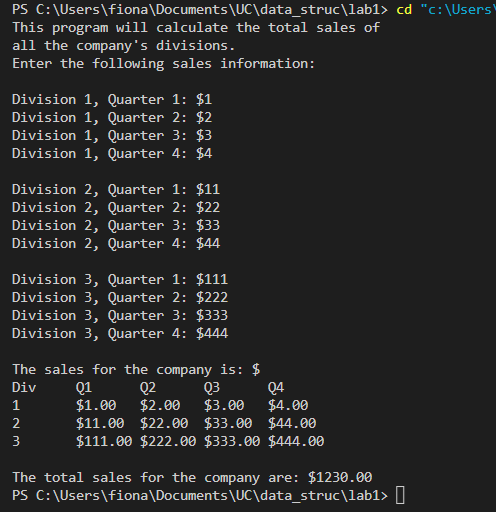


Task 2:

For task 2, we approached debugging the code by reading through it once, then running it to see the output before changing anything. Once the problems were clear, we could review the areas in the code responsible for outputting incorrect information and work backwards from there, identifying the components of each step of the process until finding the source of the issues.

In this task, the problems occurred in the lines responsible for printing information and calculating the sum of sales. The first two mistakes were small misplacement of variables.The other issue was simply a matter of the total sales not being calculated to begin with. The programmer likely just didn’t realize they typed the wrong variables in the wrong positions and forgot to calculate the sum of sales, before printing the variable for it.

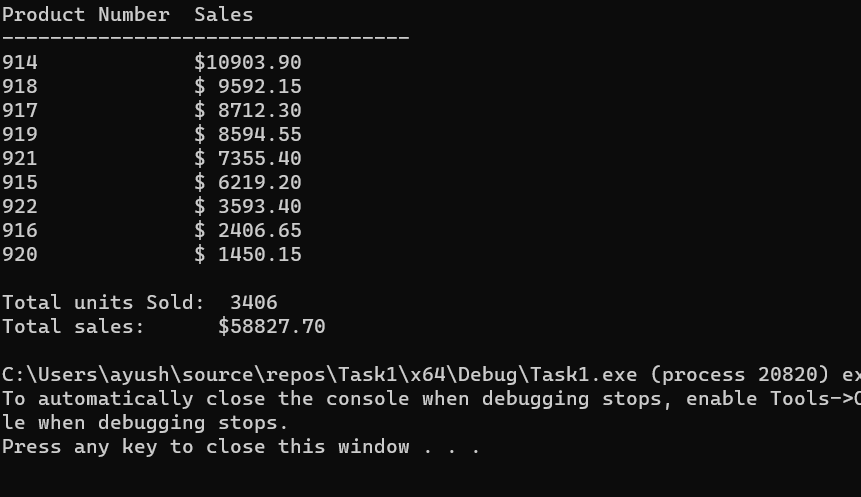
Preventing these sorts of bugs is difficult because they’re small enough that most people likely wouldn’t notice they typed the wrong thing until testing it. However, being more careful and being conscious of what each aspect of the code does as you’re writing it seems like the most effective way to avoid these issues. Using clear variable names is also good practice that can help prevent these issues. For example, if the programmer were paying close attention while writing the code, they may have realized that they were printing a constant where an incrementing value should have been.



(Task 2 output)

Task 3:

For this task we replaced the four arrays of data with a single array for product sales by creating a structure with members id, units, prices, and sale. We then adjusted the rest of the code to use the single product sales array when accessing data, rather than passing multiple separate arrays to each function. The brief bugs we encountered were when we initially implemented the struct and changed the function arguments. But once we made sure to pass the array by reference to the functions which modify the data the issues were resolved.



(Task 3 output)