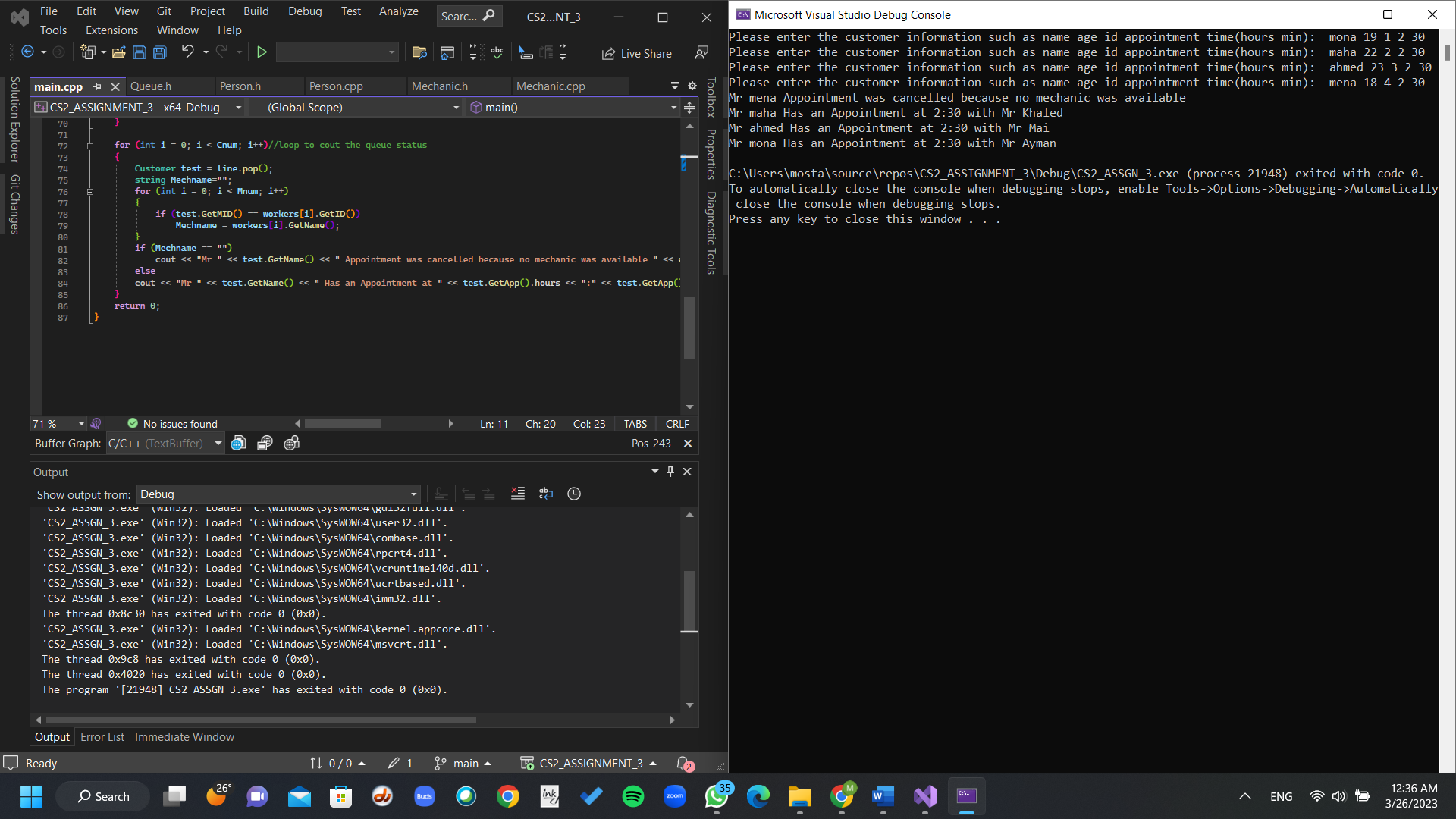
I had a problem with the queue template class as I kept getting errors when trying to use it from a .cpp and .h file. The errors would go away when I wrote the whole class in the main. I searched through the internet and found an answer to my problem on stack overflow <https://stackoverflow.com/questions/495021/why-can-templates-only-be-implemented-in-the-header-file>

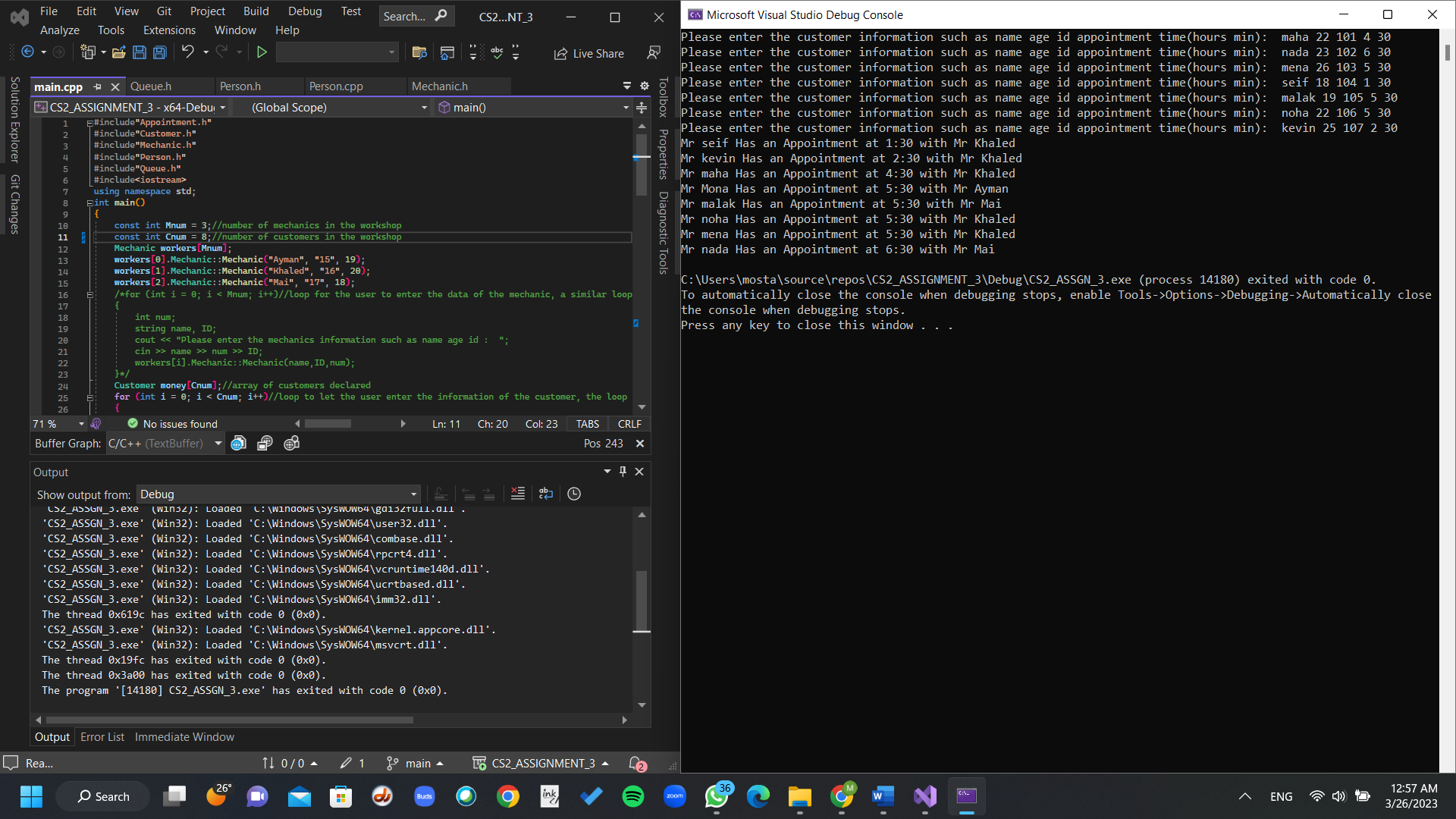
that basically said template classes should be implemented in one file on in a .h and .tpp file not .cpp

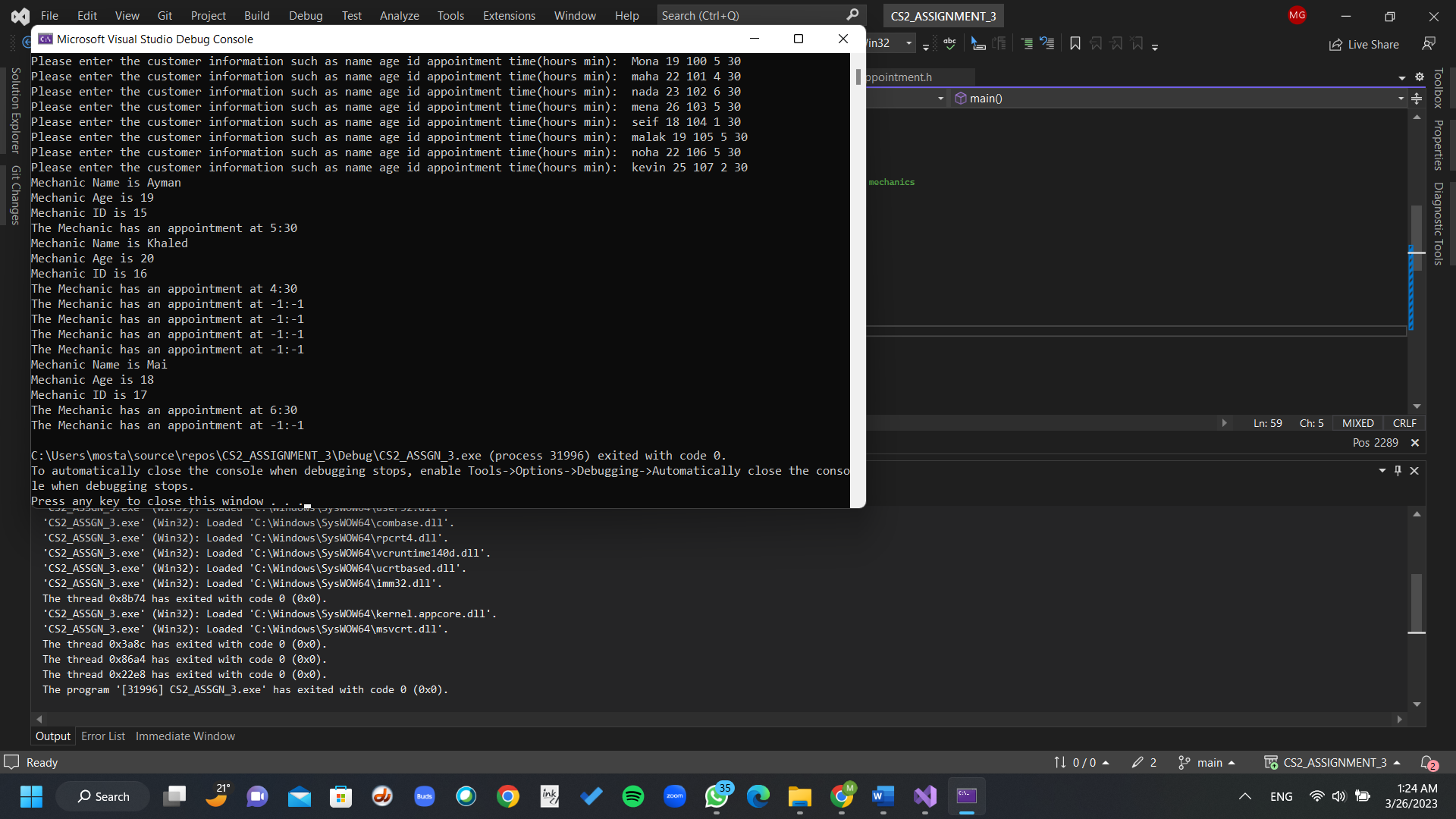
I also had a few problems with the implementation of the pop and push function because of minor errors in counters.

I made the customer class in such a way that if the appointment was set to -1:-1 and then the print info function was used it will cout that no mechanic was available to the said customer, in the main I implemented a loop that asked the user to enter the info of the customer however if no mechanic was available to said customer the appointment will be set at -1:-1 and the Mechanic Id will be set to “” which gave me two way to detect that a customer had no mechanic assigned to them because of availability example of the error is displayed down. The customers with no appointments will be displayed first because when comparing -1:-1 is technically earlier than any other appointment. I could have made the class so that if the appointment is cancelled its set at 1000:1000 which will make it appear at the end of the queue but I didn’t do that because there was no requirement for the cancelled appointments to appear last.

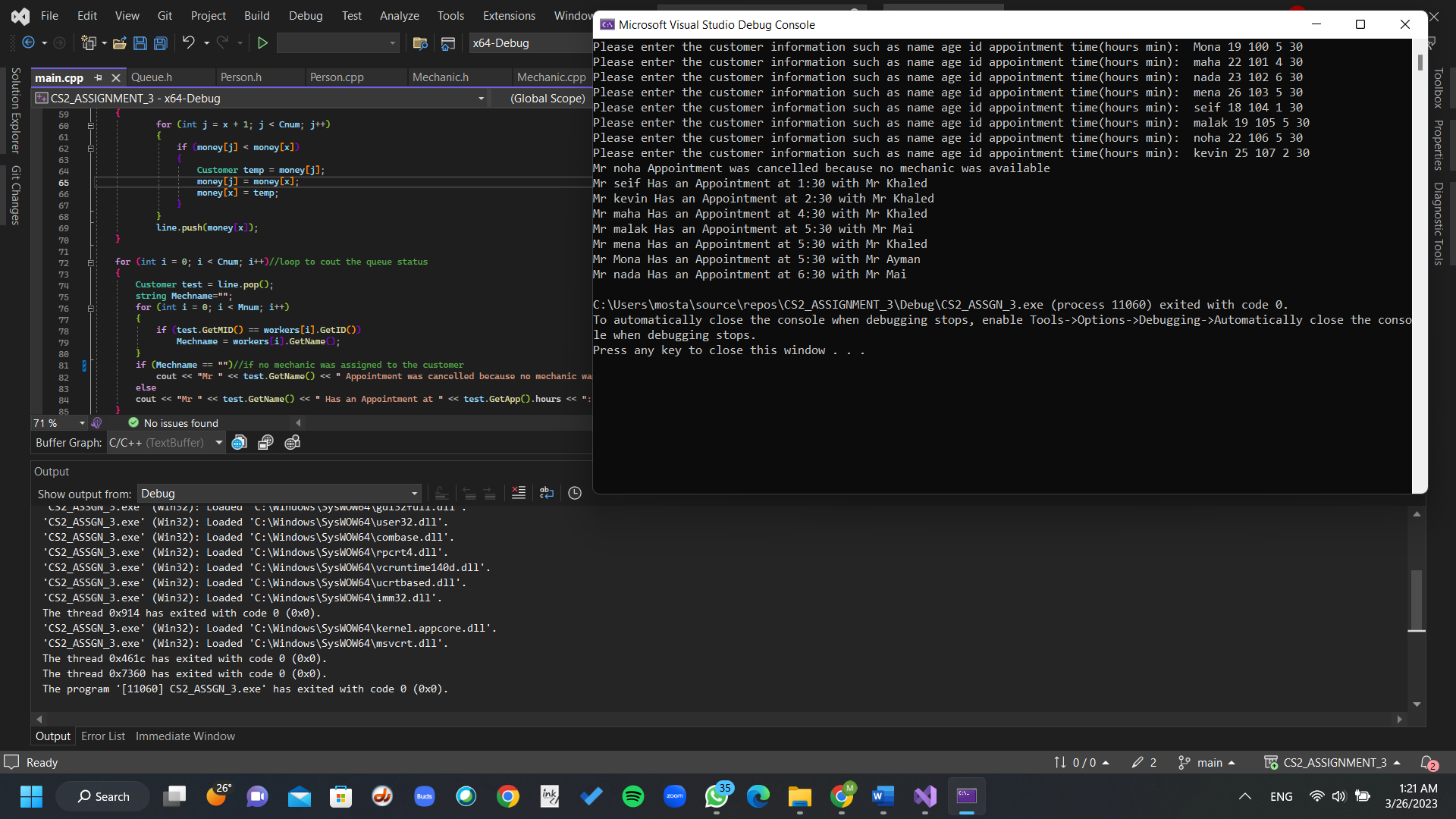


I had a problem with the program because it wouldn’t use is available from the mechanic class correctly which resulted in mechanics having multiple appointments at the same time. I used the mechanic.printinfo function which made it obvious that the problem was actually with the setApp function. Turns out I wrote this->arr[count++]; instead of this->arr[count++]=temp;

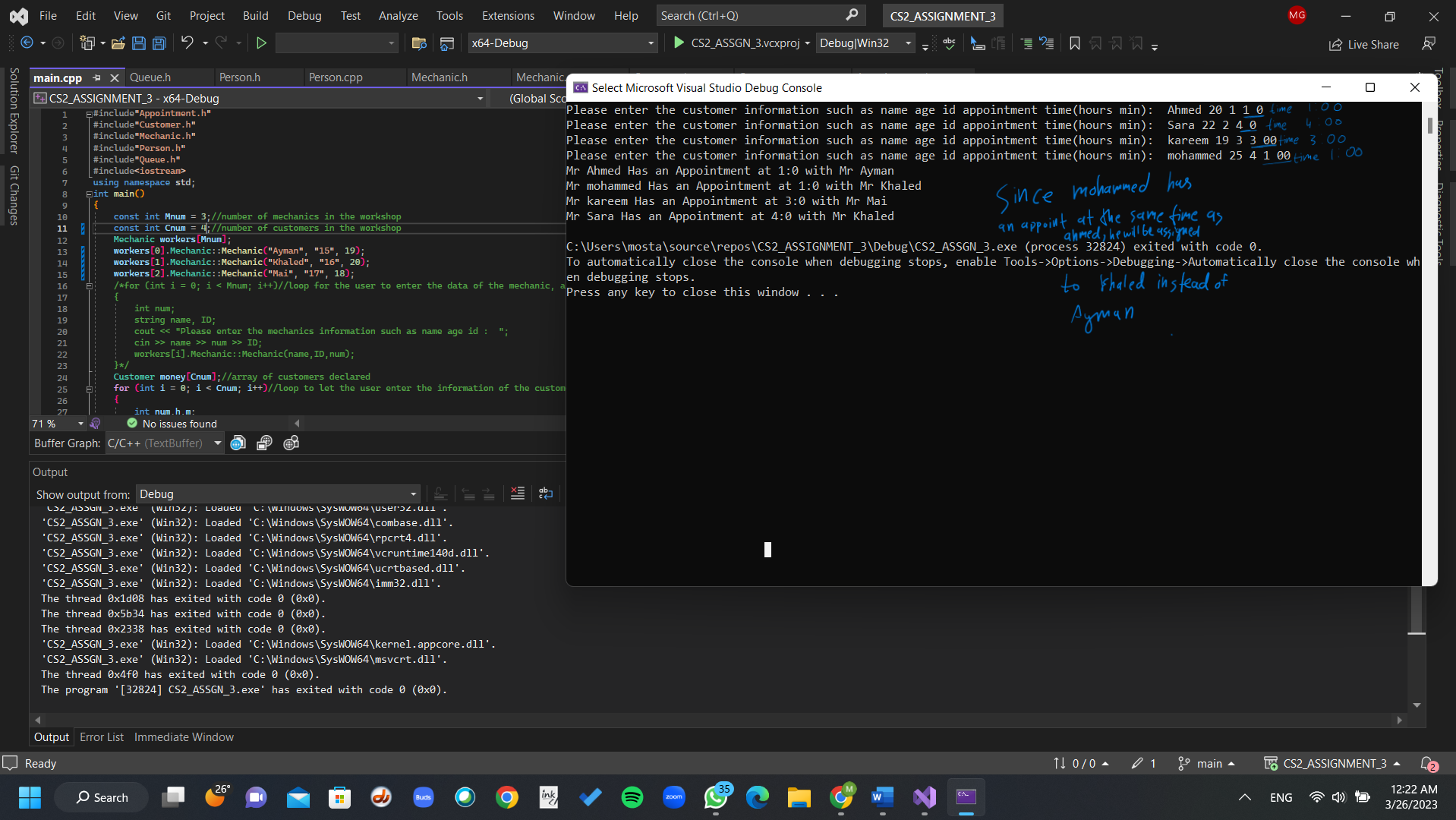




The output after fixing it :



The working program with the given inputs and correct outputs is displayed below



I used a constant Cnum to represent the number of customers and a Mnum to represent the number of Mechanics which can be changed and the program will adapt. In this case I used Cnum=4 and Mnum=3. However it can be any number

In my opinion a better implementation would be to include a pointer to a mechanic as a data member in the Customer class instead of the mechanic ID, this will make it easier to get the name of the mechanic AND the ID of the Mechanic too assigned to the said customer, it also makes more sense however that’s just my opinion. I didn’t do it because I thought I might get marks deducted because I added things that wasn’t asked.