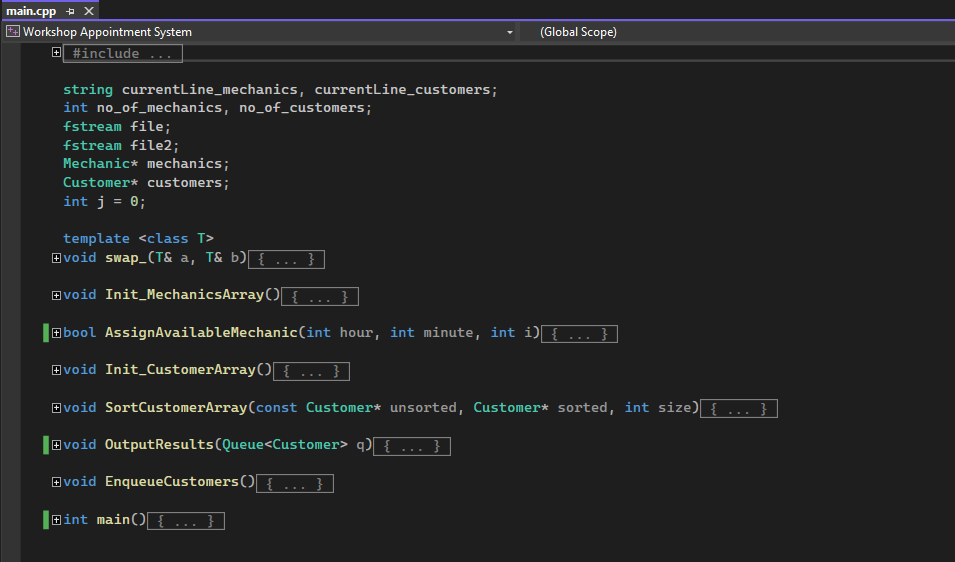
**Workshop Appointment System Report:**

**i. PROGRAM WORKFLOW**

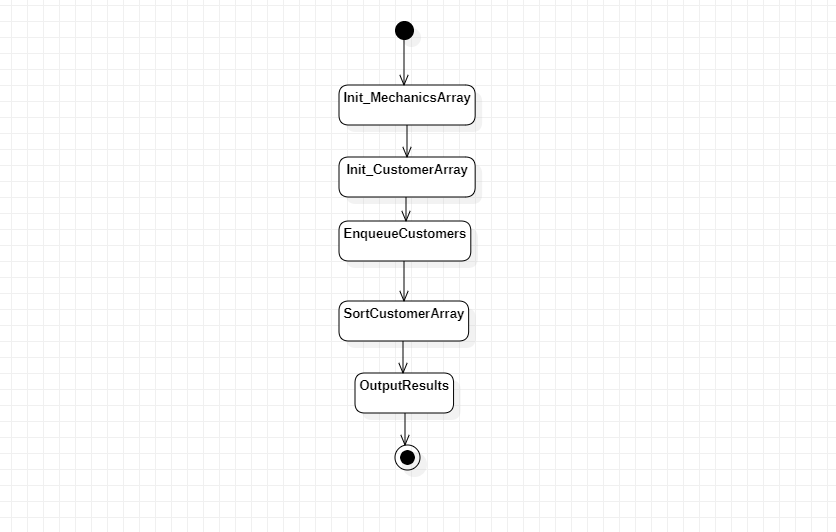
NOTE: This section of the report is adding to the already provided information in the assignment 3 documentation/guidelines.

**MAIN.CPP**

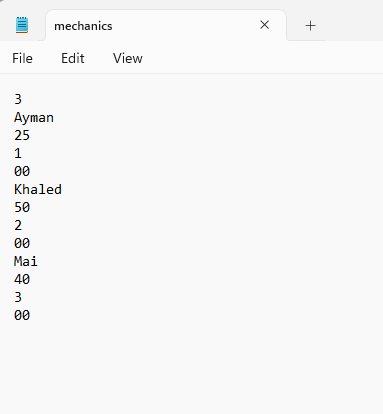
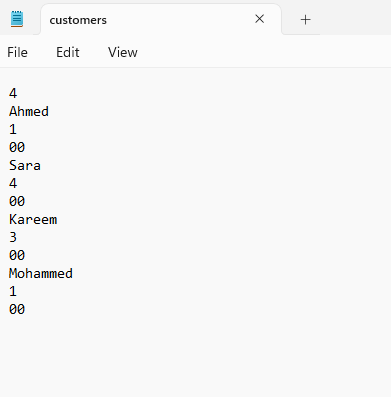


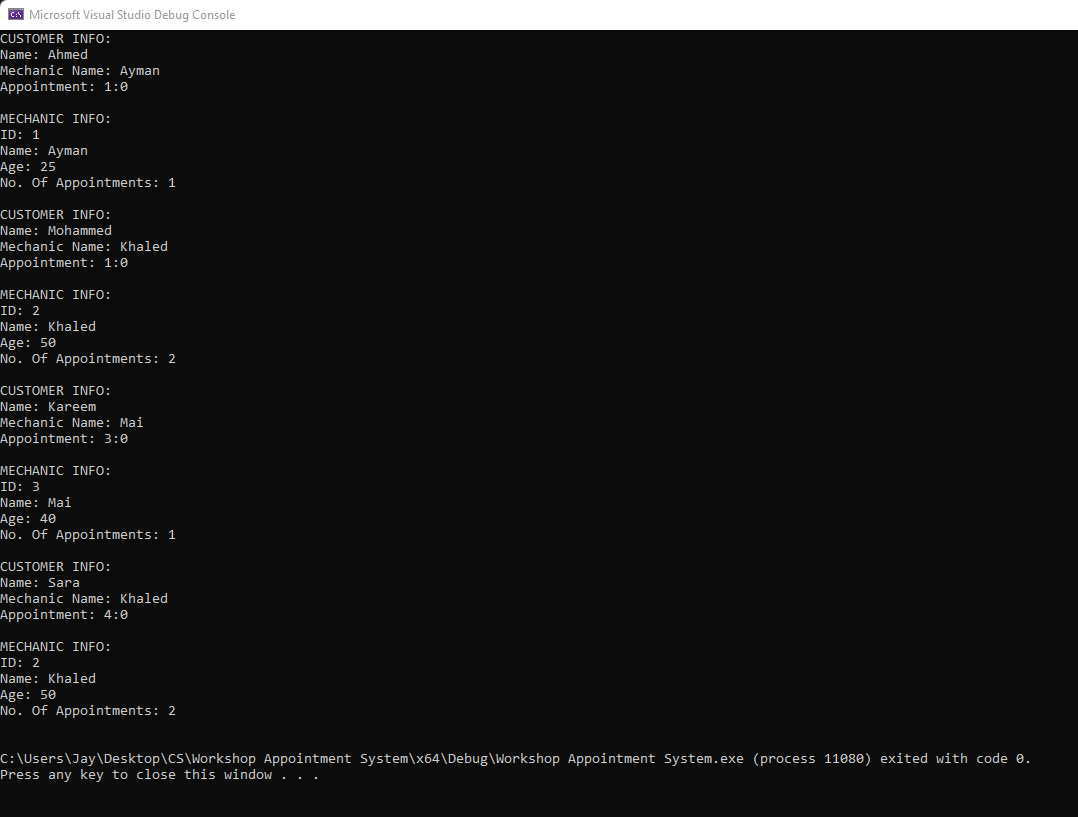
In the main section of the program, both the mechanics and customer text files containing needed data such as the number of mechanics and customers, names, ids and appointment details are opened. Then, the mechanic and customer dynamic arrays are created of size number\_of\_mechanics and number\_of\_customers which is stored in the .txt files.

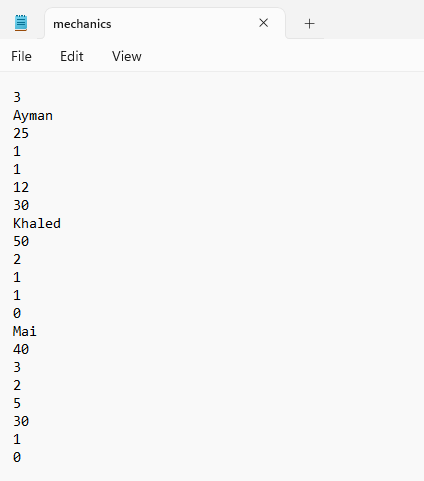
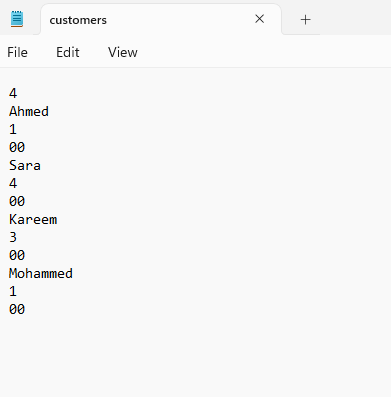
1. The Init\_MechanicsArray function is then called which traverses the text file and fills in each mechanic object’s parameters accordingly.
2. The Init\_CustomerArray function is called which traverses the customer text file and fills in each customer object’s parameters.
3. Finally, the EnqueueCustomers function is called which starts a chain of events leading to the final output. First, the queue is initialized (of size no\_of\_customers) and a new dynamic array is created which will contain the existing customers in a sorted order. Second, the SortCustomerArray function is called and the existing array of customers is sorted and placed into the sortedCustomers array. Third, each element in the sorted array is pushed into the newly created queue. Lastly, the OutputResults function is called with the queue as the argument.
4. The OutputResults function works by using a for loop and popping each customer off the queue sequentially and also printing the corresponding mechanic details by calling the .printInfo() method for each of these customers and mechanics.



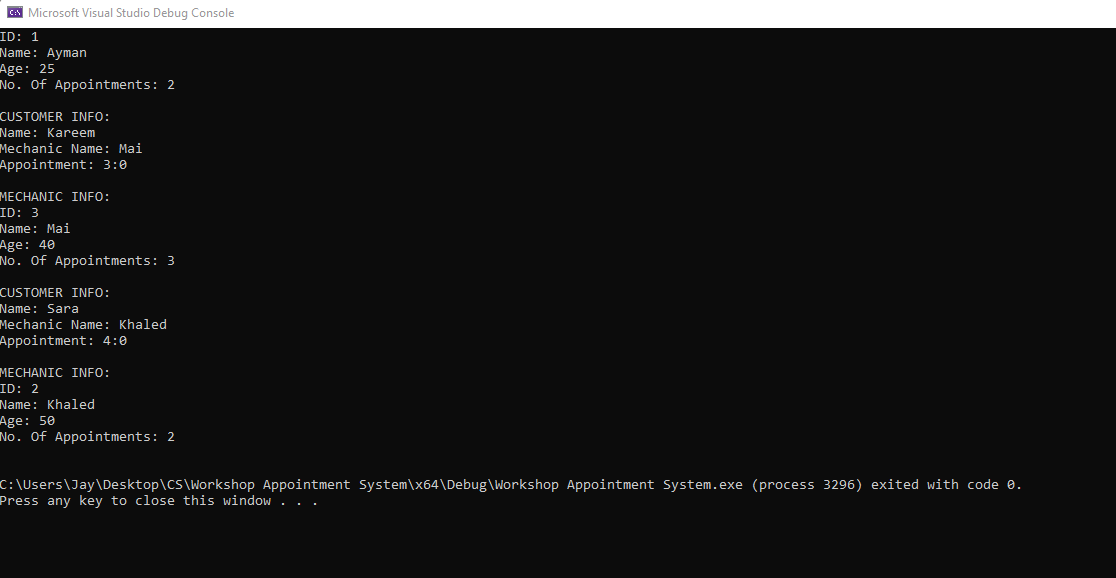
**ii. TEST CASES**

****

The first out of the two test cases was the given example in the assignment guidelines. Four customers and three mechanics. In this test case, all the mechanics have no previously existing appointments. All customers manage to receive their appointment. The output is as follows:



The second test case used the exact same customer file and the same mechanics, however, this time the mechanics already have pre-existing appointments in the .txt file and not all customers manage to get their desired appointments. The output is as follows:



**iii. ADDITIONAL NOTES**

This program includes all the wanted attributes for each class asked for in the assignment guidelines, however, it also includes many additional methods and attributes that were implemented to make the output process much simpler. E.g, the MechanicName and noAppointment attributes in the Customer class and their corresponding getters and setters.