

CS-2255 - Computer Science II

FINAL PROJECT

Group 6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Student Name | TroyID | HustID | Phone |
| 1. | Ho Gia Huy | 1669908 | 20218155 |  |
| 2. | Nguyen Huy Hoang | 1624664 | 20218147 |  |
| 3. | Vu Tien Dung | 1624659 | 20218120 |  |

**Table of Contents:**

1. [Problem](file:///C:\Users\khoin\Downloads\Project1-Group5-CS256.docx#_Toc149847392) 
   * 1. 1. Issues.
     2. *2.* Requirements specification:
     3. 3. Ultilization.
     4. 4. Outcome
2. [Solutions:](file:///C:\Users\khoin\Downloads\Project1-Group5-CS256.docx#_Toc149847399) 
   * 1. 1. Entity Relationship Diagram:
     2. 2. Software design.
     3. 3. Application diagram
3. **Problems:**
4. *Issues:*

Installing the program allows verifying the following sorting algorithms:

- Selection Sort

- Merge Sort

- Bubble Sort

1. *Requirements specification:*

Building program menus from the console allows users to:

5. Import a .txt file containing a series of numbers that need to be sorted – The required number of elements is no less than 10,000.

6. Choose 1 of 3 algorithms to sort

7. The user will be asked to save the sorted results as a file name provided by the user and view algorithm run result information: execution time

8. Compare the execution time of 3 algorithms, the input needs to change by 10,000, 100,000 and 1,000,000 elements to compare the performance of each algorithm

Students will set system time starting – system time ending when the algorithm has finished to calculate the running time.

1. *Utilization:*

This software can be used to compare the complexity of 3 sorting algorithm: selection sort, merge sort and bubble sort and calculate the time that given the result when sorting 10,000, 100,000 and 1,000,000 elememts.

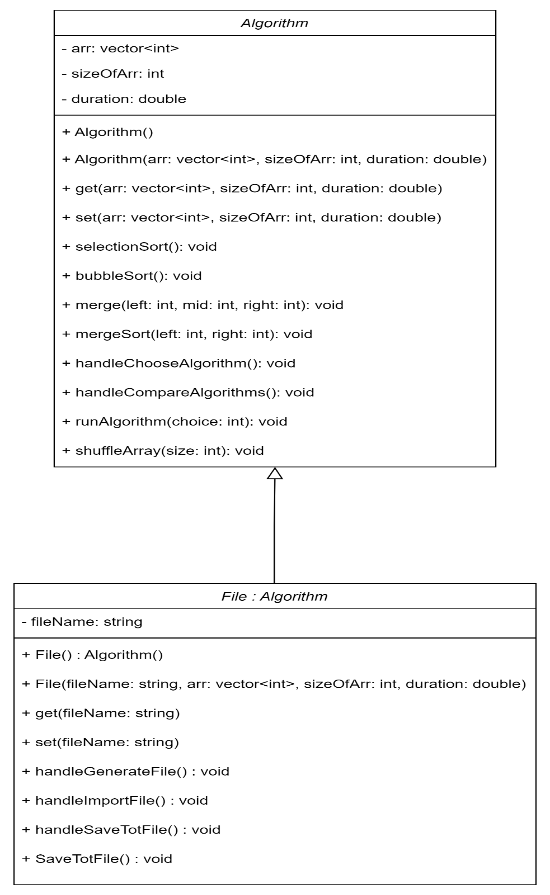
1. *Outcome*

The software helps user to have easier understanding of sorting algorithm and how to read the algorithm, time complexity, and it should be used to optimize time and efficiency for applying to advanced software.

1. **Solution:**
2. *Entity Relationship Diagram:*

In this software, we have two class relationship:

* The first class is Algorithm with 3 fields, 2 constructors and 10 methods. The second class is File extends from class Algorithm. The File class has 1 new fields, 2 constructors and 6 methods.



Picture 1.1 Entity Relationship of Algorithm and File.

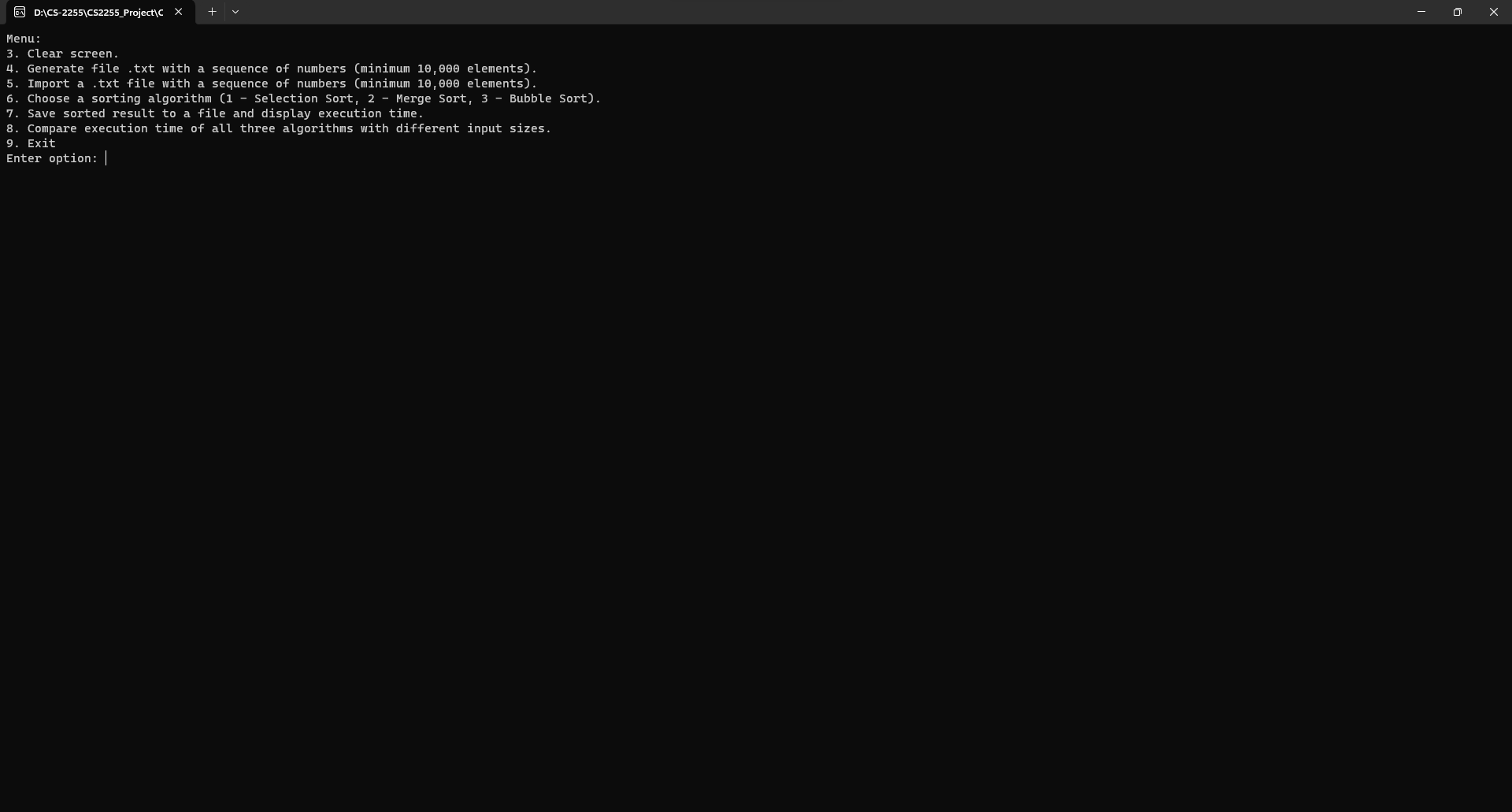
1. *Software design:*

In this software, we have two class:

* + The first is Algorithm. In this class, we have 3 fields and 8 methods:
    - Fields: The first fields is arr, with type vector of integer. It contains numbers of elements needed to sort in this program. The second is sizeOfArr, which is the number of element of arr. The third fields is duration. It is the time sorting all elements of arr of 3 algorithm.
    - Methods: The first is selectionSort. The second is bubbleSort. The third and the fourth are merge and mergeSort. These methods use sorting algorithm to sort the elements in arr. The fifth is handleChooseAlgorithm, which lets user choose 1 of 3 algorithm to sort. The sixth is handleCompareAlgorithm, which compares the execution time of 3 algorithms with 10,000, 100,000 and 1,000,000 elements. The seventh is runAlgorithm, which execute the algorithm: 1 with selection sort, 2 with merge sort and 3 with bubble sort. The last is shuffleArray, which shuffle arr using the Fisher-Yates shuffle algorithm.
  + The second is File, which extends from class Algorithm. In this class, we have 1 new fields and 4 new methods:
    - The only fields in this class is fileName, which is the name of file needed to generate, export or import.
    - Methods: The first is handleGenerateFile, which generates a .txt file with no less than 10,000 integers elements. The second is handleImportFile, which imports a .txt file with 10,000 minimum elements to the software and push to arr in the Algorithm class. The third is saveToFile, which creates a .txt file and save to it. The last is handleSaveToFile, which handle to saveToFile methods and display execution time the algorithm use to sort in the first class (duration).

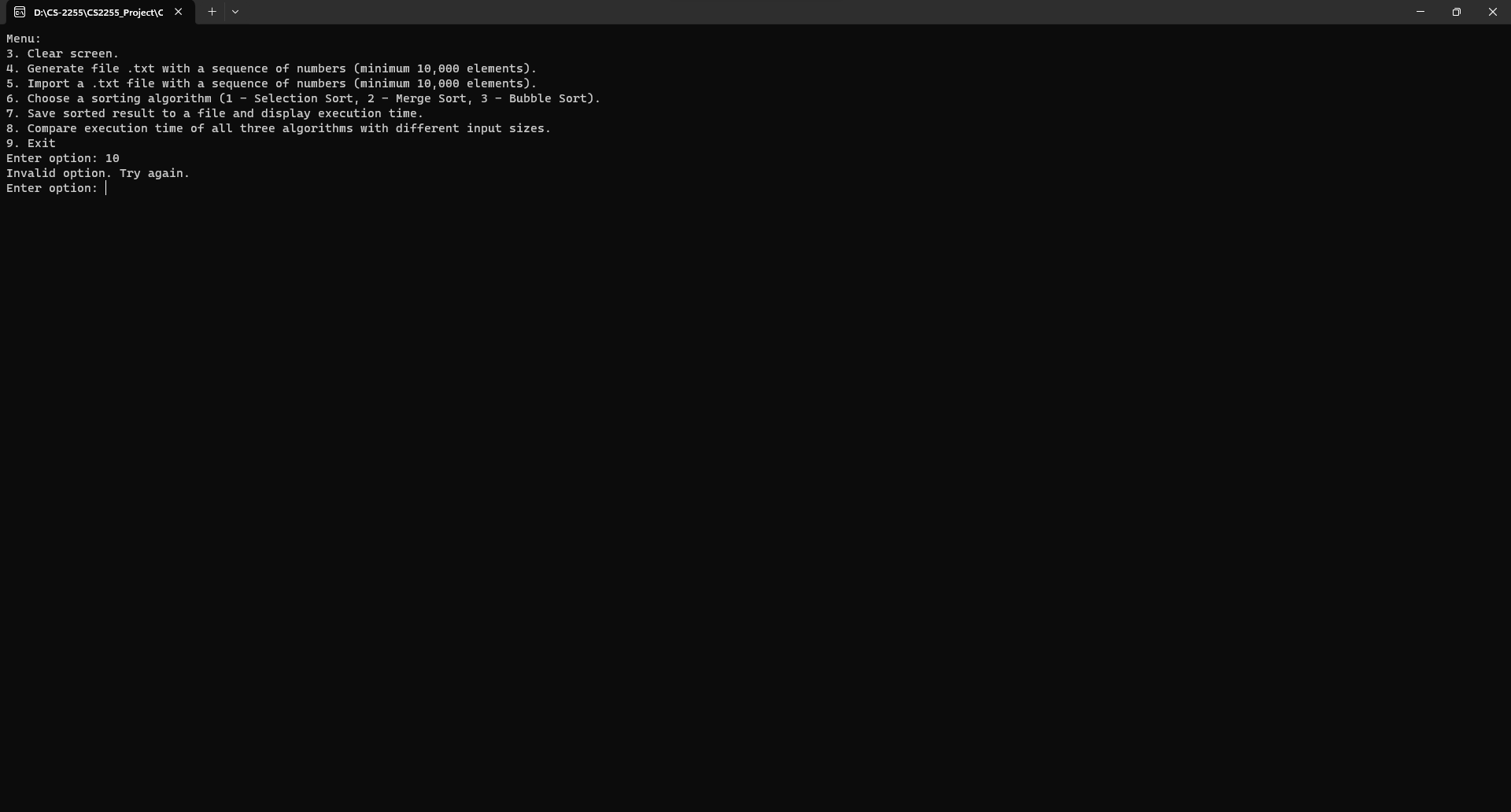
In addition, there are menu function, which display menu for users and main function, which execute the program, let user choose option and execute the program for users.

1. *Application diagram*

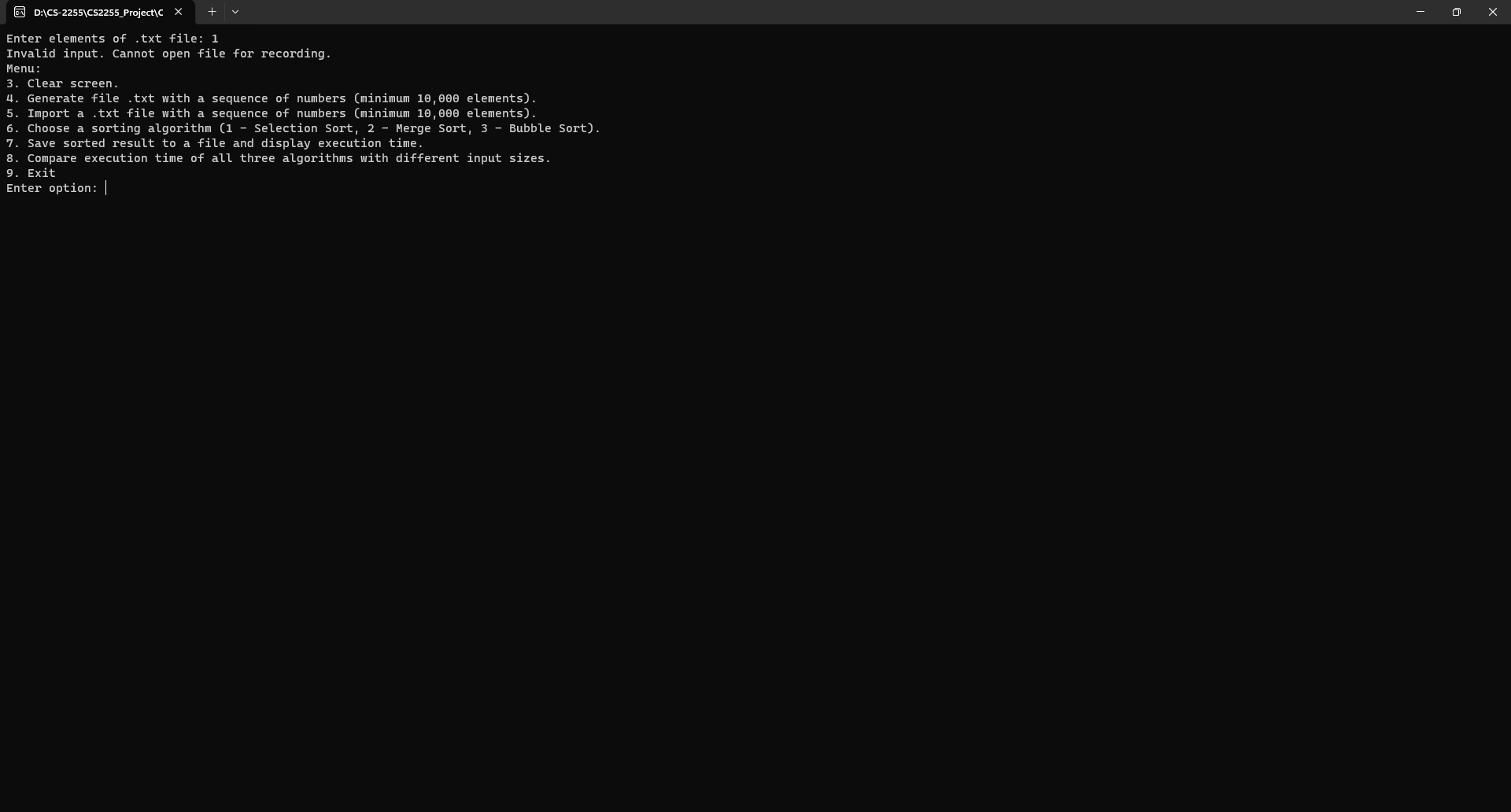


Picture 3.1 Start the program

When user starts the program or enters 3 to clear screen, it requires user to enter a number from 3 to 9 to execute the program. If user enters an invalid input, the program requires users to enter again.

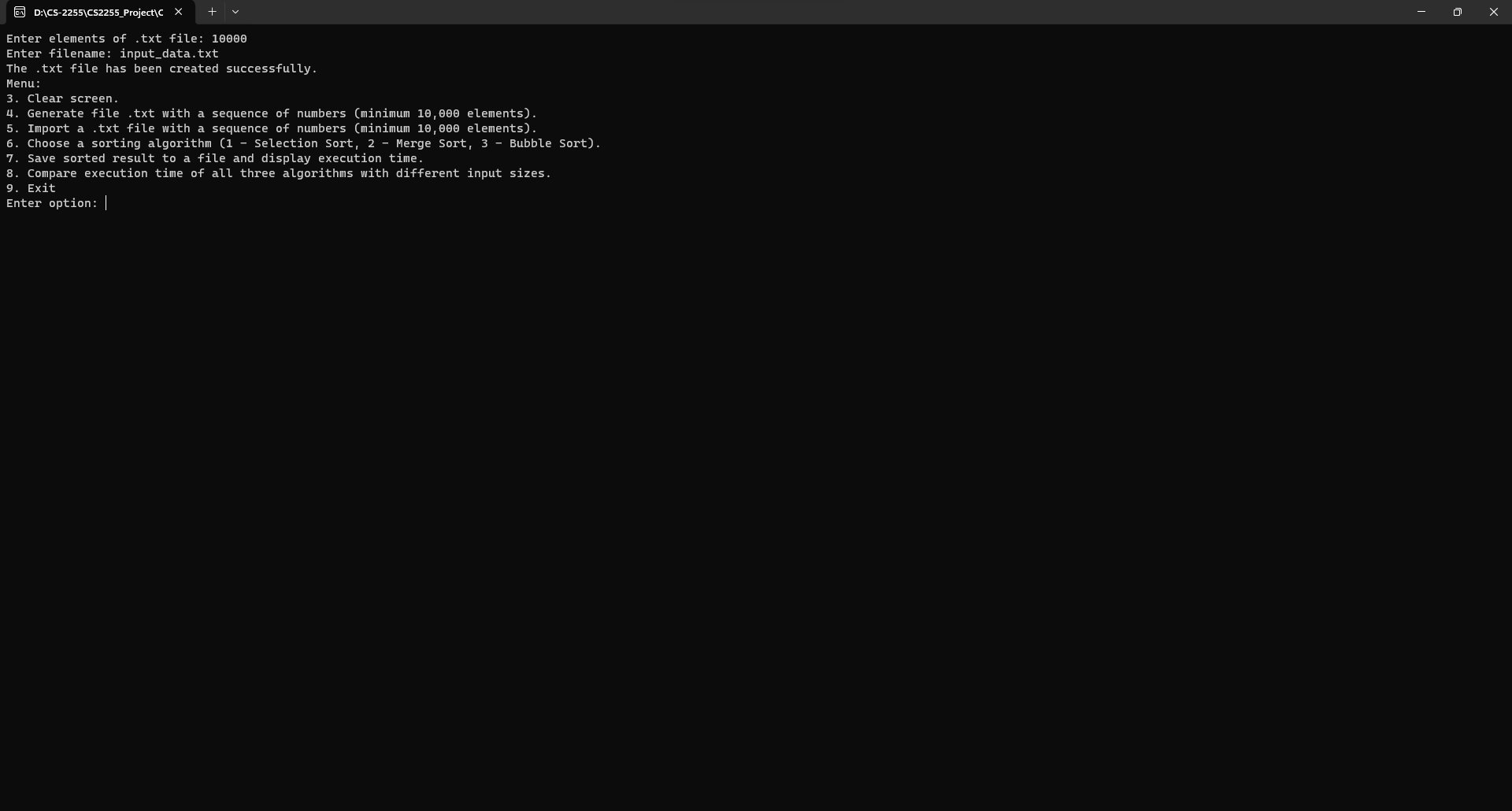


Picture 3.2 When user enters an invalid input in the menu

When user enters 4, the program requires user to enter the number of integer elements of the .txt file. The minimum number of integer elements is 10,000. If user enters an invalid input, the program shows “Invalid input. Cannot open file for recording.” and back to the menu. 

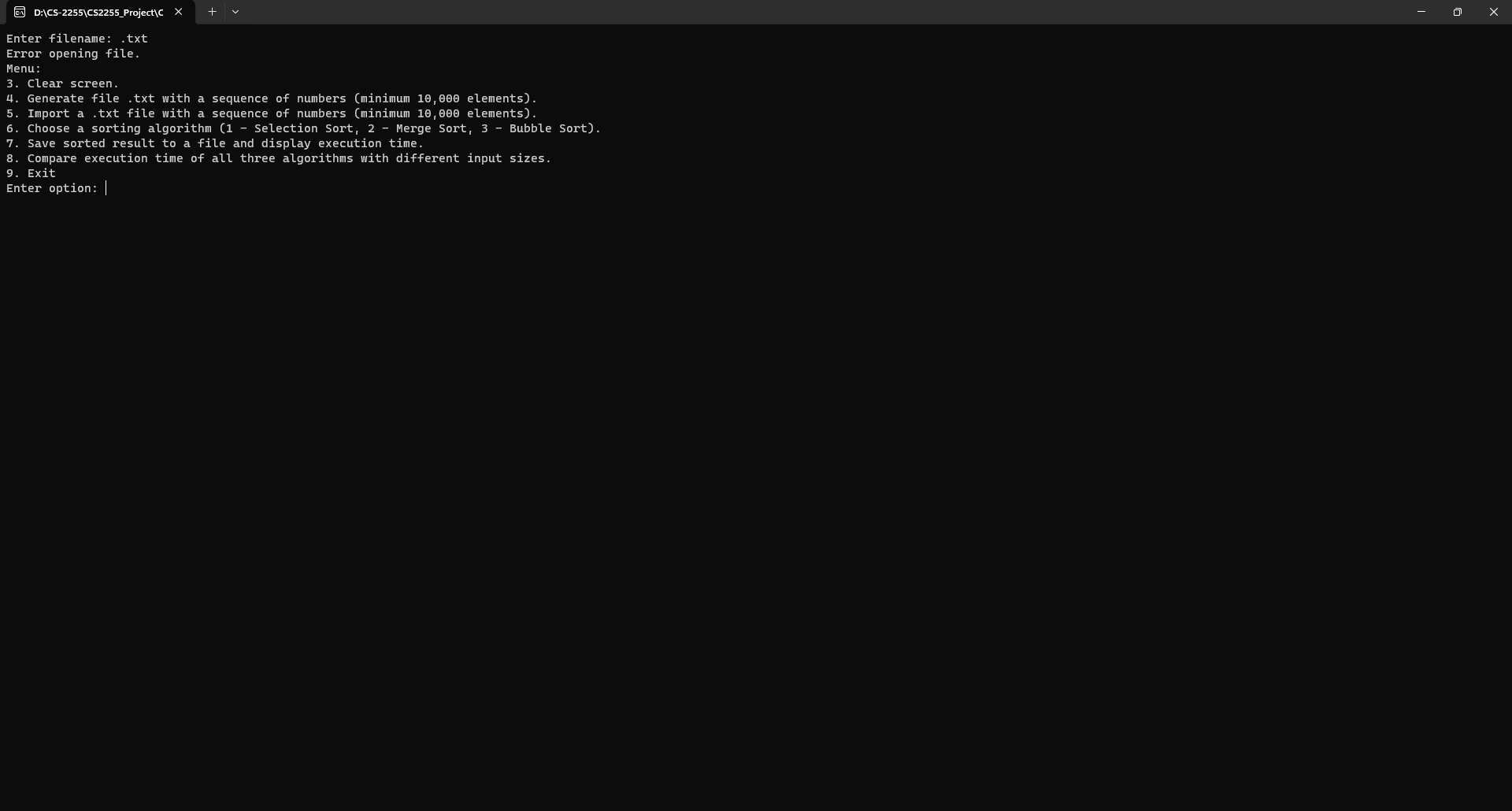
Picture 3.3 When user enters an invalid input in case 4

If user enters a valid input, the program requires user to enter the name of file. Then, the program create a file with the name user entered and save the data having random integer elements. This number of this elements in this file is the number when user enters in the first step.



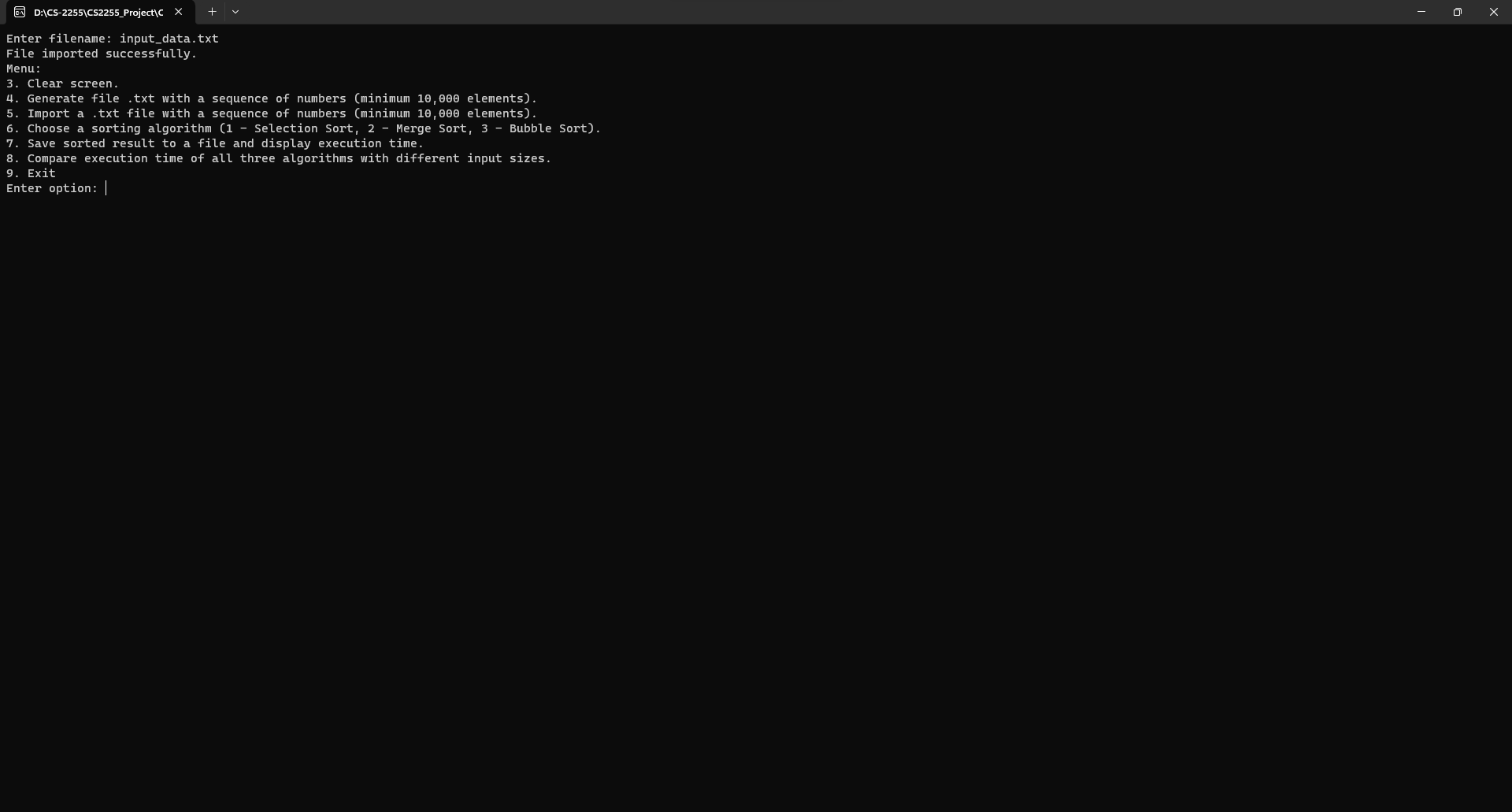
Picture 3.4 When user enter an valid input in case 4 and enter a file name after that.

When user enters 4, the program requires user to enter the .txt file needed to import. If user enters an invalid file (enters a file which is not found or enters a file which having less than 10,000 numbers of elements), the program shows “Error opening file.” and back to the menu.



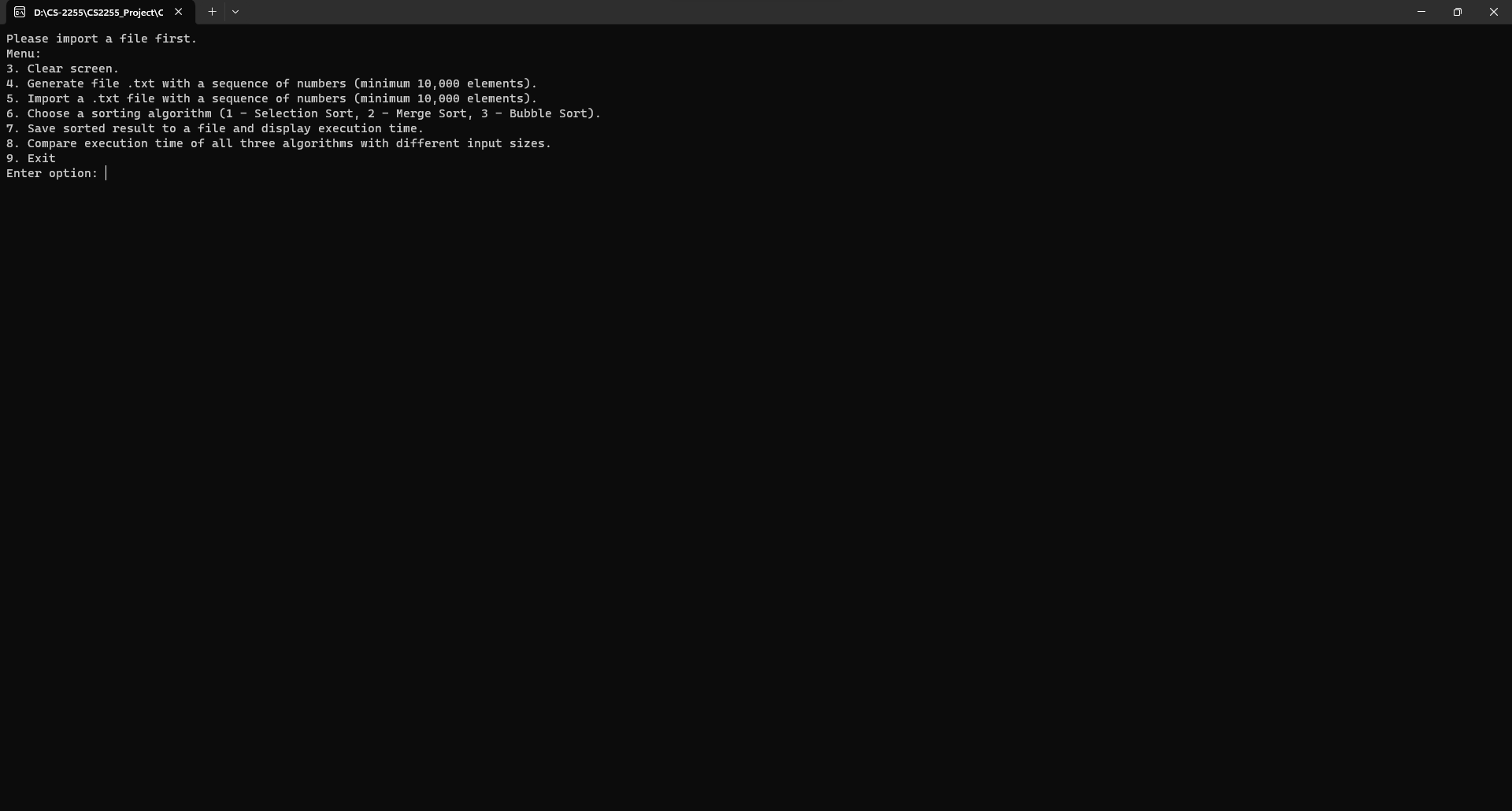
Picture 3.5 When user enters an invalid file in case 5

If user enters a valid file, the program shows “File imported successfully”. The array of minimum 10,000 integer elements is pushed from this file, and this array can be sorted in case 6.



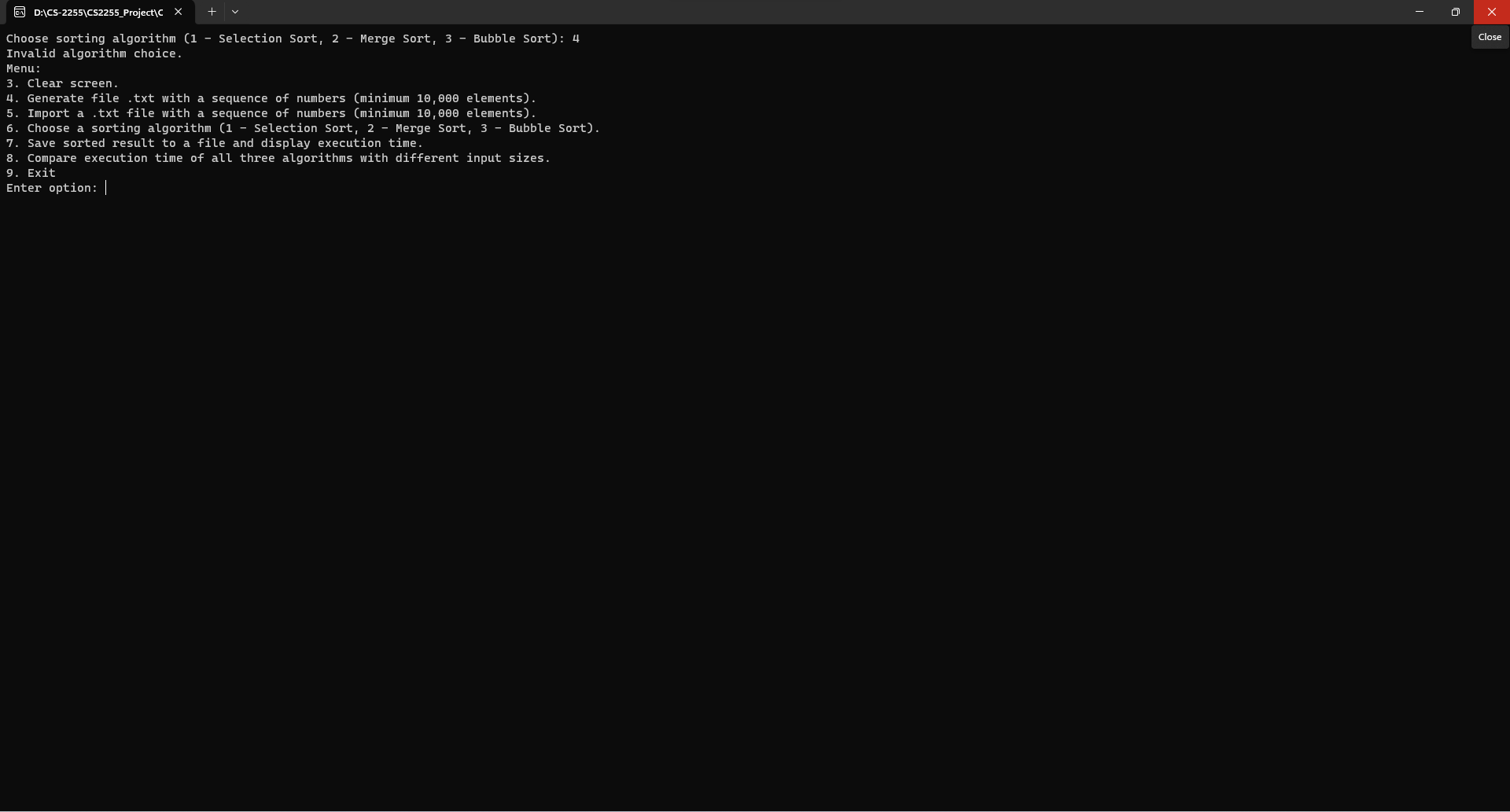
Picture 3.6 When user enters a valid file in case 5

When user enters 6, if user did not import file in case 5, the program requires user to import file first and back to the menu.



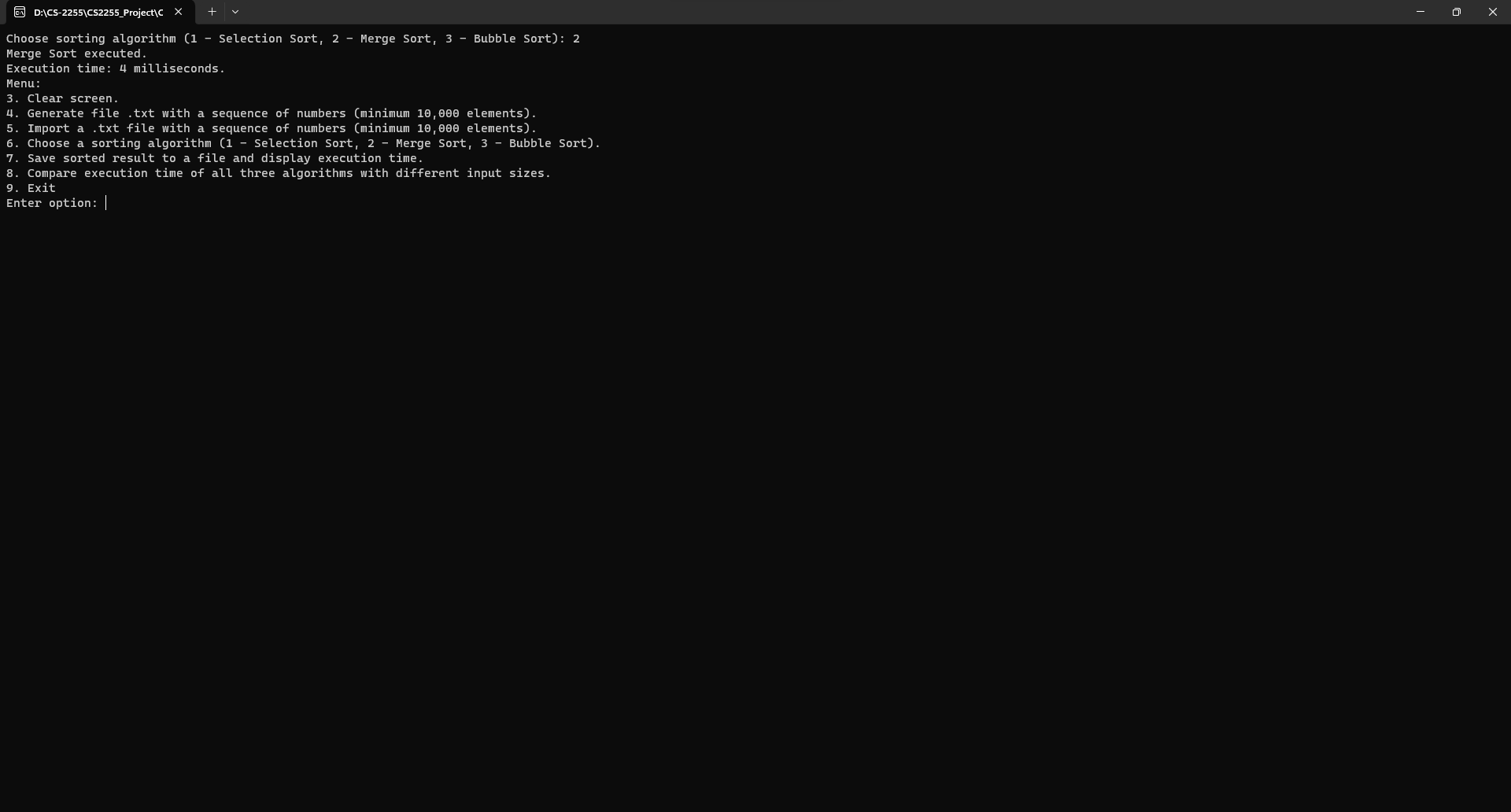
Picture 3.7 When user enters 6 but does not import file in case 5

If imported file in case 5, the program requires user to choose sorting algorithm: 1 for selection sort, 2 for merge sort, and 3 for bubble sort. If user enters an invalid input, the program show “Invalid algorithm choice” and back to the menu.



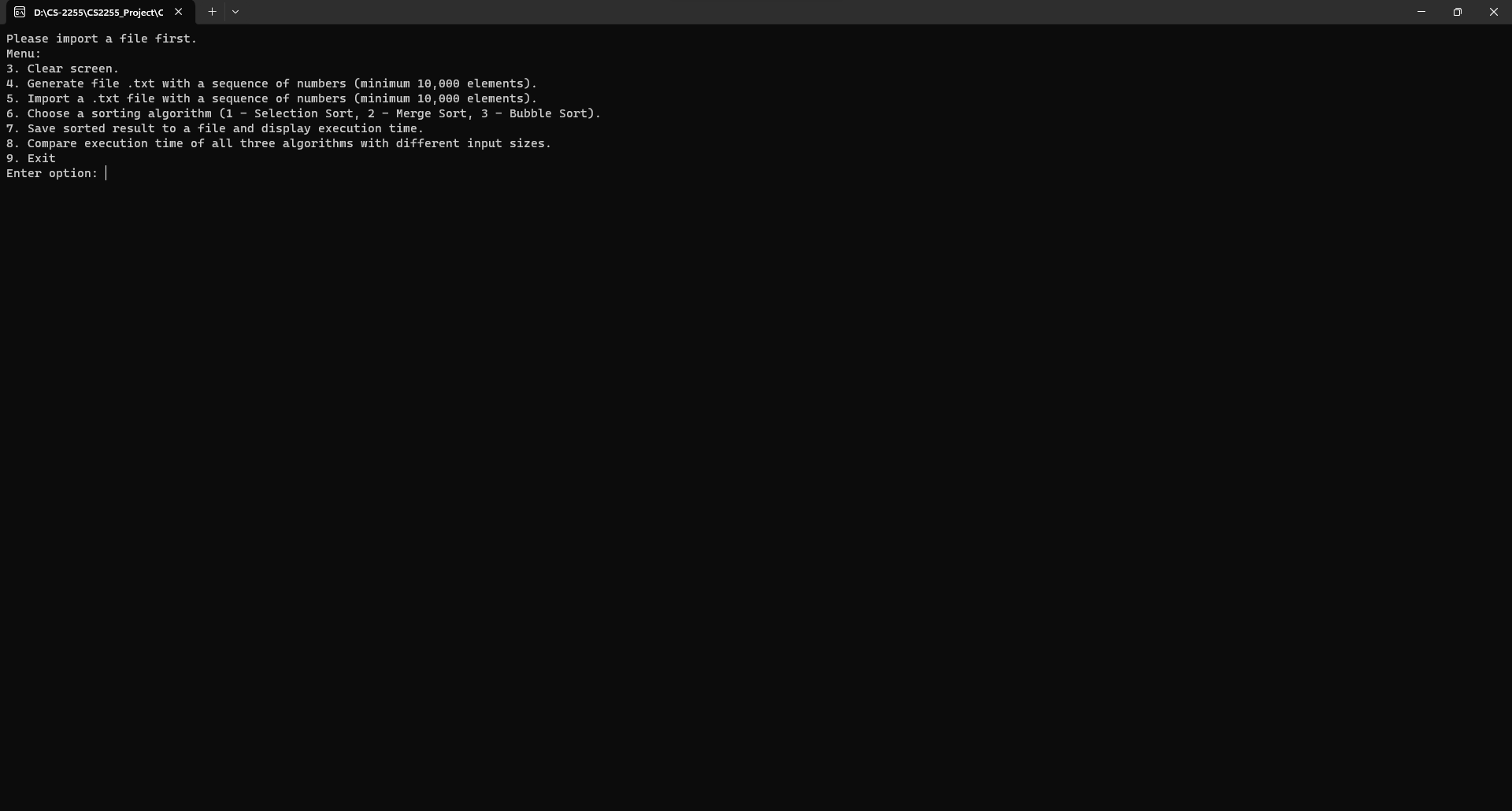
Picture 3.8 When user enters an invalid input in case 6

If user enters an invalid input, the program show that the sort algorithm is executed and display the execution time for user. The array imported in case 5 is sorted and it can be saved to a file in case 7.



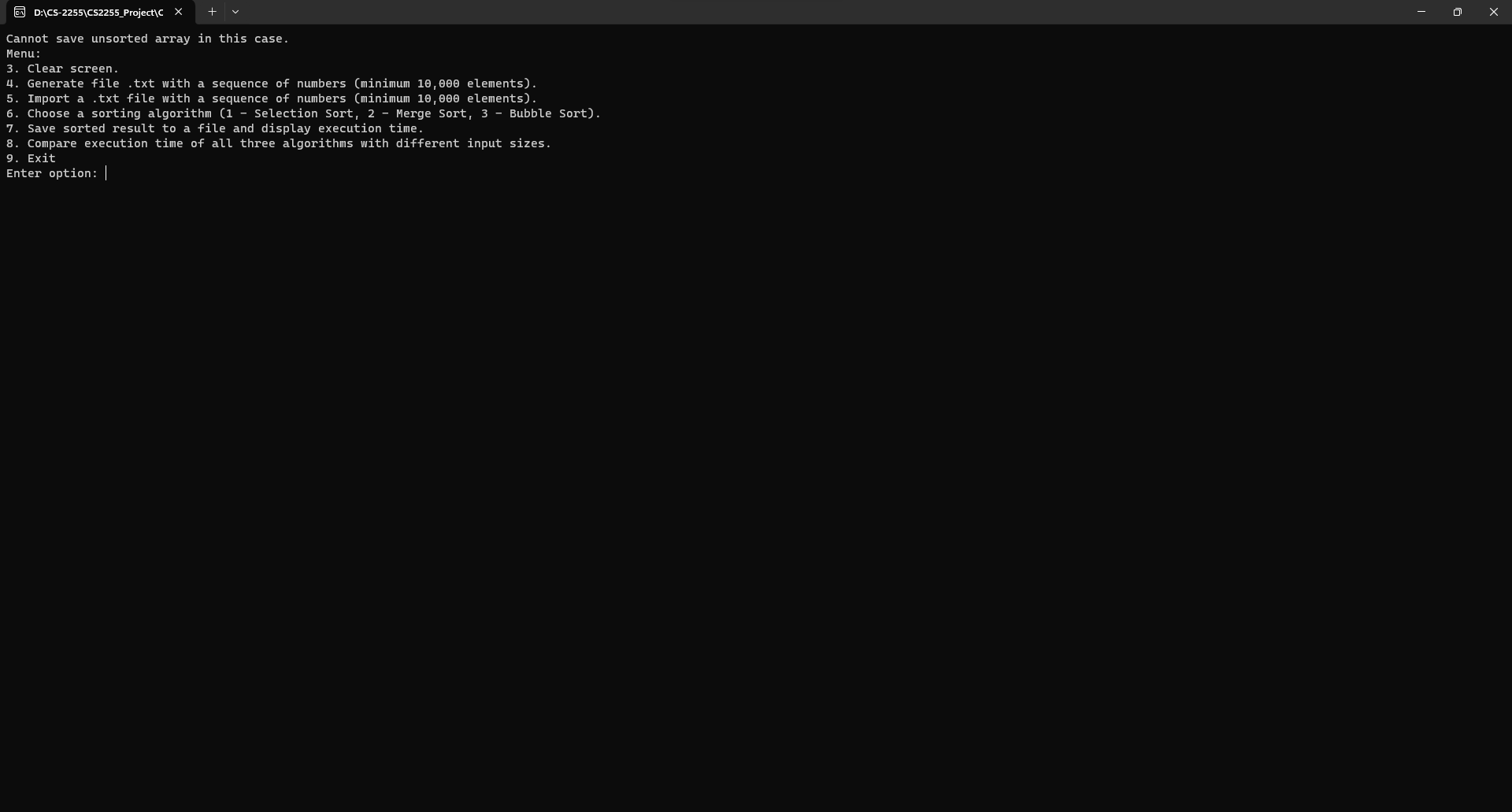
Picture 3.9 When user enters an valid input in case 6

When user enters 7, if user did not import file in case 5, the program requires user to import file first and back to the menu.



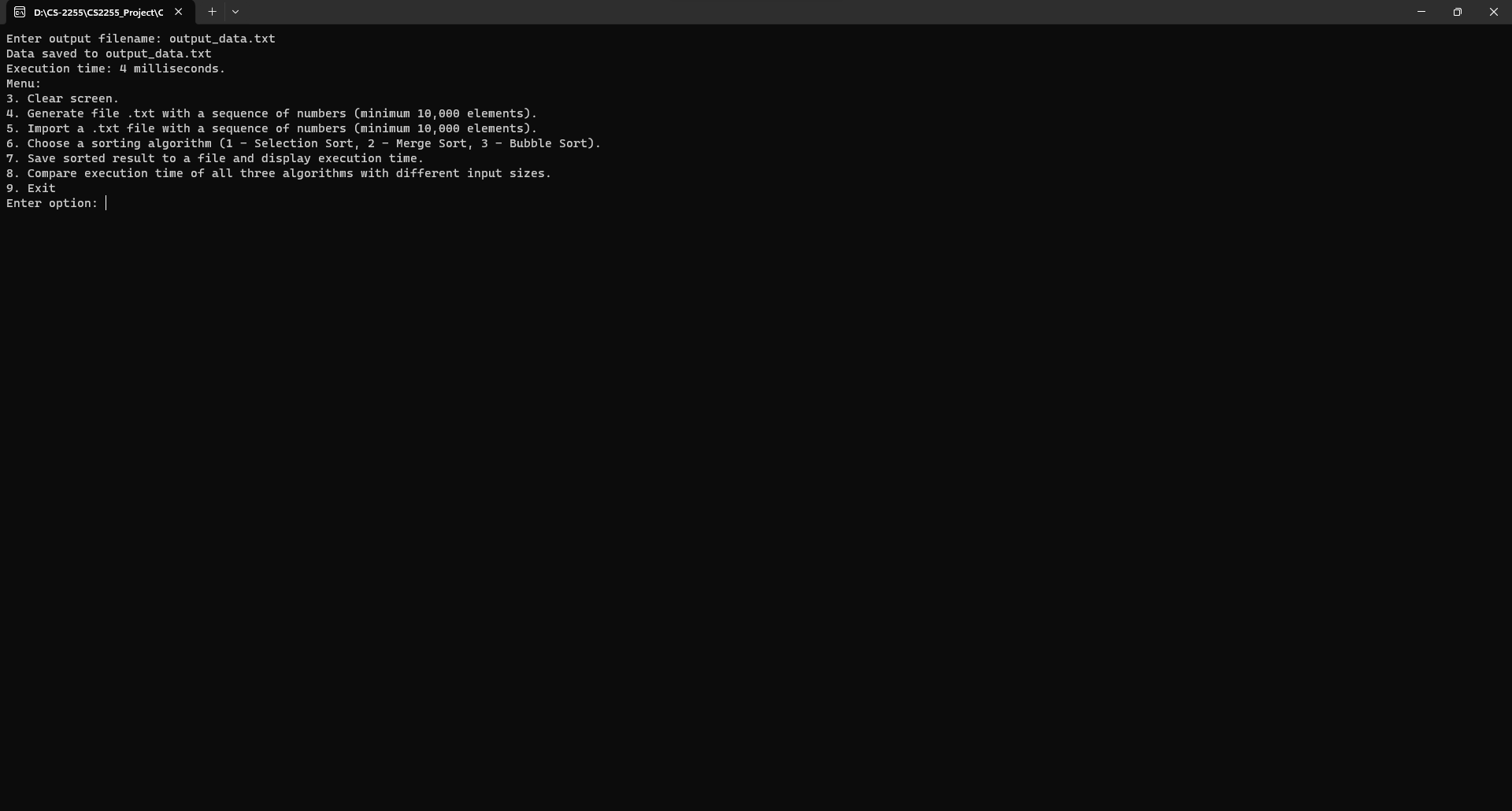
Picture 3.10 When user enters 7 but not import file in case 5

If imported file in case 5, if user did not sort in case 6 before, the program shows “Cannot save unsorted array in this case. ” and back to the menu.



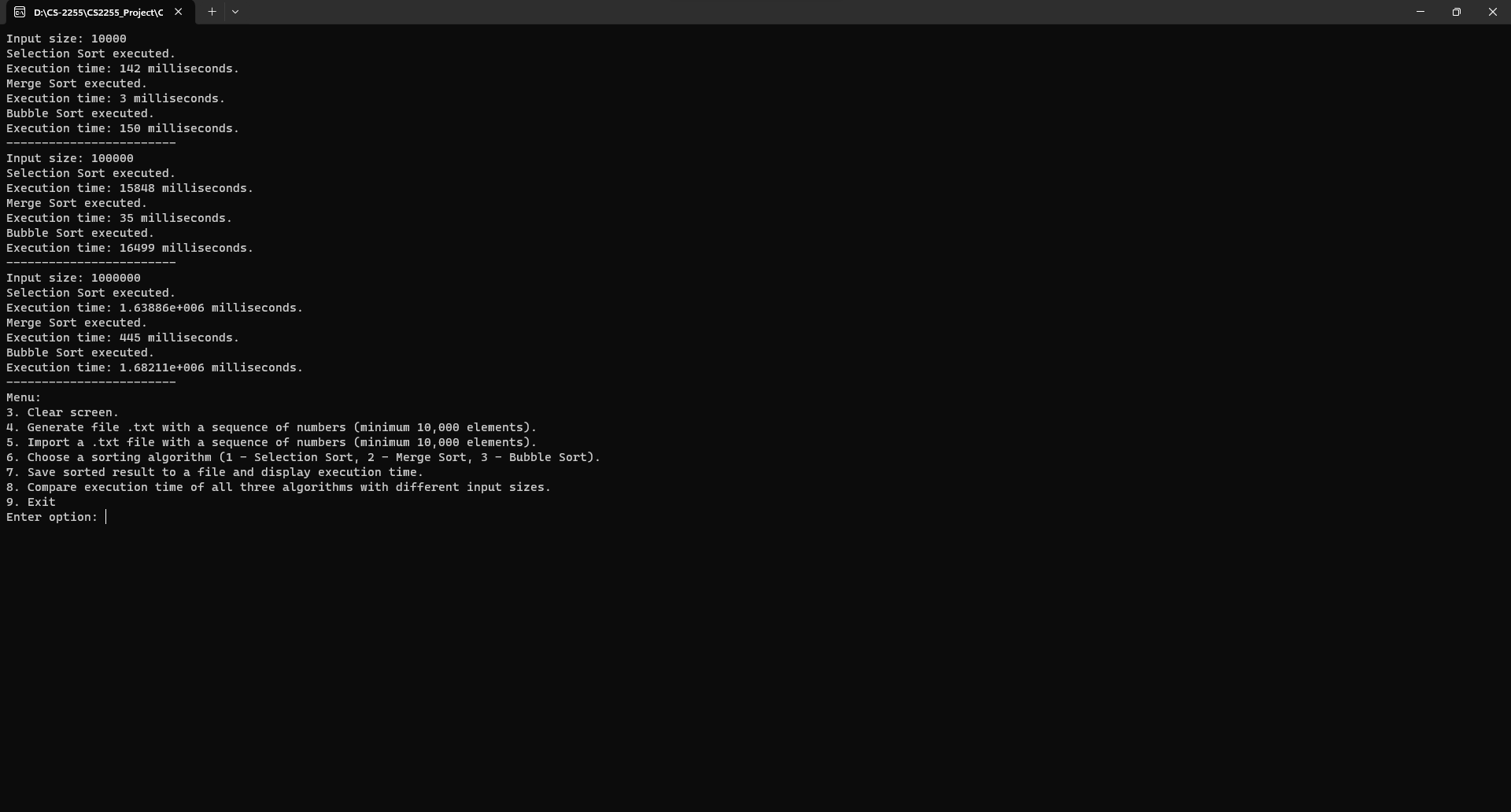
Picture 3.11 When user enters 7 after import file in case 5 but not sort data in case 6

If imported file in case 5, if user sorted in case 6 before, the program requires user to enter the name of file needed to save. Then, the program create a file with the name user entered and save the data having all elements sorted in case 6.



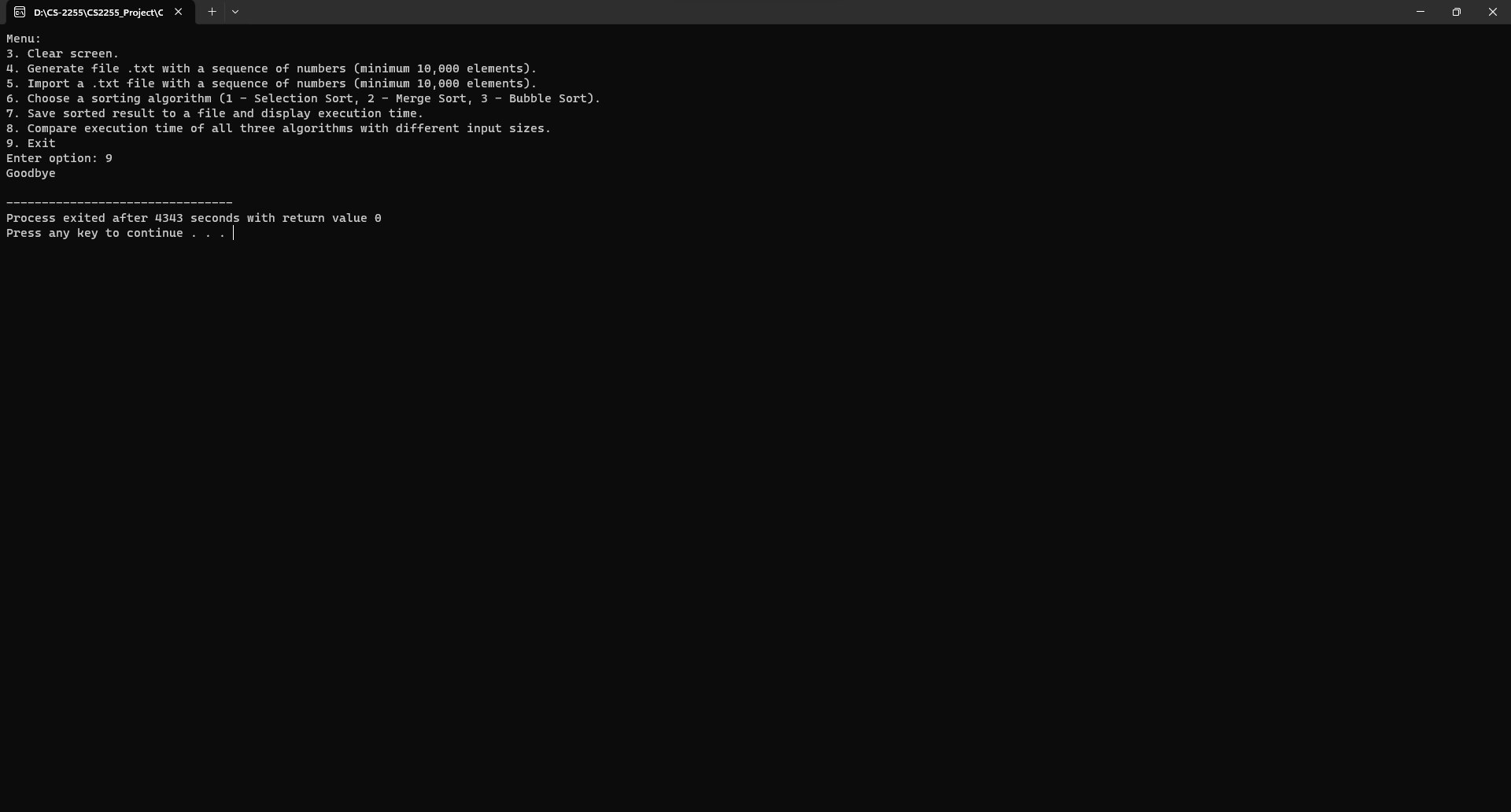
Picture 3.12 When user enters 7 after import file in case 5 and sort data in case 6

When user enters 8, the program executes 3 sorting algorithm to compare the time sorting array. Input size changes: the first has 10,000; the second has 100,000 elements; the third has 1,000,000 elements. The program shows the time executing 3 sorting algorithm after sorting them in three case and display time execution for user to compare the complexity of them.



Picture 3.13 When user enters 8

Finally, when user enters 9, the program will exit.



Picture 3.14 When user enters 9