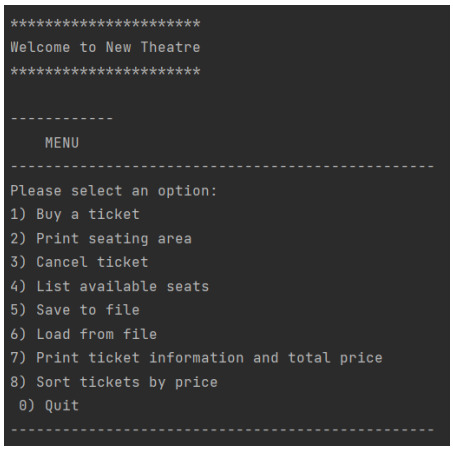
Student name: Janitha Prabodha
Student ID: w1953789 | 20220212
Tutorial group (day, time, and tutor) – info is in your timetable:
Group E, Monday, 01.30 PM, Mr. Lakshan Costa

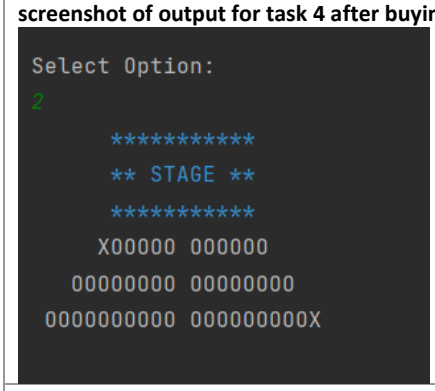
Fill the following table. For test input, expected output and output obtained, add as many cases as you have tested.

See an example of a completed form in Blackboard.

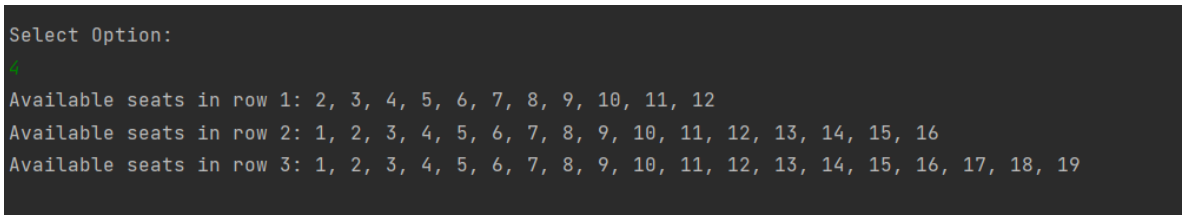
Task	Self-assessment (select one)	Test input	Expected output	Output obtained	Comments
1	☒Fully implemented and working ☐Partially implemented ☐Not attempted	1: User runs the program.	1: "Welcome to new theatre" + Menu Displayed	1: "Welcome to new theatre" + Menu Displayed	The Welcome message is displayed
2	☒Fully implemented and working ☐Partially implemented ☐Not attempted	2: Input from the user > Inputs a Invalid number	2: "Select option" + "Invalid option"	2: "Select Option" + "Invalid Option"	Handling Invalid User Input:
Insert screenshot of your menu here including the welcome message:  ********** Welcome to New Theatre ********** --- MENU --- Please select an option: 1) Buy a ticket 2) Print seating area 3) Cancel ticket 4) List available seats 5) Save to file 6) Load from file 7) Print ticket information and total price 8) Sort tickets by price 0) Quit ---					
3	☒Fully implemented and working ☐Partially implemented ☐Not attempted	Input from the user > Inputs a Non-integer	"Integer Required. Please select an option from the menu"	"Integer Required. Please select an option from the menu"	Handling Non-Integer User Input:
4	☒Fully implemented and working ☐Partially implemented ☐Not attempted	Menu Option 1 > Row 1 > Seat 1 + Menu Option 1 > Row 3 > Seat 20 + Option 4	"Ticket Purchased Successfully! " + Ticket Purchased Successfully! + Available seats in row 1: 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 Available seats in row 2: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 Available seats in row 3:	"Ticket Purchased Successfully! " + Ticket Purchased Successfully! + Available seats in row 1: 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 Available seats in row 2: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 Available seats in row 3:	First seat of the first row and the last seat of the last row are successfully purchased and Available seats are show accordingly

			1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19	

screenshot of output for task 4 after buying a ticket for row 1, seat 1 and row 3, seat 20 :



screenshot of output for task 6 after buying a ticket for row 1, seat 1 and row 3, seat 20 :



5	☒Fully implemented and working ☐Partially implemented ☐Not attempted	Menu Option 1> Row 1 > Seat 1 > (Case 1) Name: Saman Surname: Kumara Email: Saman@email.com Price: 10$	"Ticket Purchased Successfully! " + Person Object created and data added + Ticket Object created and data added	"Ticket Purchased Successfully! " + Person Object created and data added + Ticket Object created and data added	Adding a Person information to the ticket
6	☒Fully implemented and working ☐Partially implemented ☐Not attempted	(Case 1) + Menu option 7	Print the user data, + Total Price Of tickets	Print the user data, + Total Price Of tickets	Printing the ticket information.

Screenshot of the output of Print ticket information (Option 7) with multiple user inputs

7	☒Fully implemented and working ☐Partially implemented ☐Not attempted	User selects the option 1. Enter: Row 1 + Seat 1	"Enter Row number " +	"Enter Row number " +	Purchasing a ticket

			"Enter seat number"	"Enter seat number"	
8	☒Fully implemented and working ☐Partially implemented ☐Not attempted	Input Option 2 >	Displays the seating area	Displays the seating area	Checking if the seat is booked in the seating area
9	☒Fully implemented and working ☐Partially implemented ☐Not attempted	Case 1 : Input Option 3 > Row number 1 > Seat Number 1 >	"Ticket cancelled successfully"	"Ticket cancelled successfully"	Cancelling a ticket and removing the ticket object from the array
10	☒Fully implemented and working ☐Partially implemented ☐Not attempted	(Case 1) + Input Option 2	"Seating area Displayed." + "Booked seats are represented with X"	"Seating area Displayed." + "Booked seats are represented with X"	Checking the seating area after cancelling the ticket.
11	☒Fully implemented and working ☐Partially implemented ☐Not attempted	(case 2) + Input Option > seat 1 row1> option 1 > seat > row 2 > seat 1 >Option 1> row 3 > seat 1	Message "Seat booked Successfully" every time	Message "Seat booked Successfully" every time	Buying a ticket from each row
12	☒Fully implemented and working ☐Partially implemented ☐Not attempted	(case 2) + Menu Option 2	"(Case) Booked seats are represented with X"	"(Case) Booked seats are represented with X"	Checking if the seats from each row is represented buy a "X" in the seating area
13	☒Fully implemented and working ☐Partially implemented ☐Not attempted	(Case 2) + Menu Option 1 > Row 1 > Seat 1	"Seat is occupied. Please try again"	"Seat is occupied. Please try again"	User again tries to buy the Row 1 seat 1: The seat is occupied. message is displayed
14	☒Fully implemented and working ☐Partially implemented ☐Not attempted	(Case 2) + Menu option 3 > Row 1 > Seat 1> option 1 > row 2 > seat 1 > Option 1> row 3 > seat 1	All the selected seats are shown as Not occupied	All the selected seats are shown as Not occupied	Tring to cancel multiple seat which are not occupied
15	☒Fully implemented and working ☐Partially implemented ☐Not attempted	Menu Option 1 > Input row 4	"This row does not exist, Please Select (1 - 3)"	This row does not exist, Please Select (1 - 3)	select option 1 and enter value greater than 3 for the Row number, Error Displayed
16	☒Fully implemented and working ☐Partially implemented ☐Not attempted	Menu Option 1 > Input Row 1 > Seat 13	"This Seat does not exist, Please Select (1 - 12)"	"This row does not exist, Please Select (1 - 3)"	Checking the first row Range is 1 - 12

17	☒Fully implemented and working ☐Partially implemented ☐Not attempted	Menu Option 1 > Input Row 2 > Seat 17	"This Seat does not exist, Please Select (1 - 16)"	"This row does not exist, Please Select (1 - 16)"	Checking the second row Range is (1 - 16)
18	☒Fully implemented and working ☐Partially implemented ☐Not attempted	Menu Option 1 > Input Row 3 > Seat 21	"This row does not exist, Please Select (1 - 20)"	"This row does not exist, Please Select (1 - 20)"	Checking the Third row Range is (1 -20)
20	☒Fully implemented and working ☐Partially implemented ☐Not attempted	(case 1) + Menu option 8	"Sorting Tickets..." + "Tickets sorted Successfully! "	"Sorting Tickets..." + "Tickets sorted Successfully! "	Adding multiple user inputs and sorting according to price

Before Sorting the ticket information: **After Sorting:**

```
Select Option:
7
Row 1, Seat 1 purchased by
SamanKumara (Saman@email.com) for 10.0  £.

Row 1, Seat 2 purchased by
JohnSmith (jhon@email.com) for 20.0  £.

Row 1, Seat 3 purchased by
anneMarie (Anna@email.com) for 15.0  £.

-------------------
Total price: 45.0 $
-------------------
```

```
Select Option:
7
Row 1, Seat 1 purchased by
SamanKumara (Saman@email.com) for 10.0  £.

Row 1, Seat 3 purchased by
anneMarie (Anna@email.com) for 15.0  £.

Row 1, Seat 2 purchased by
JohnSmith (jhon@email.com) for 20.0  £.

-------------------
Total price: 45.0 $
-------------------
```

21	☒Fully implemented and working ☐Partially implemented ☐Not attempted	Menu Option 3 > Row 2 > seat 2 (When not occupied)	"Seat is NOT Occupied, Please Try again."	"Seat is NOT Occupied, Please Try again."	Cancelling a seat which is NOT occupied
22	☒Fully implemented and working ☐Partially implemented ☐Not attempted	Menu Option 4 > (Without booking any seats)	All seats are shown as available	All seats are shown as available	User selects Option 4 without booking any seats
23	☒Fully implemented and working ☐Partially implemented ☐Not attempted	(Case 2) + Menu Option 5	The seating data is saved to a file	The seating data is saved to a file	Save to file

Screenshot of the seats after saving into a file

24	☒Fully implemented and working ☐Partially implemented ☐Not attempted	(Case 2) + Menu Option 6> Menu Option 2 >	Seating area is displayed according to the loaded data	Seating area is displayed according to the loaded data	Loading a data file and Printing the seating area

Screenshot of the seating area after loading data from a file

```
      ***********
      ** STAGE **
      ***********
     XOOOOO OOOOOO
   OXOOOOXX XOOOOOOO
 OOXOOOOOOO OOOOOOOOX
```

26	☒Fully implemented and working ☐Partially implemented ☐Not attempted	Option 8 (Without entering any values)	"No values to be sorted"	"No values to be sorted"	Sorting tickets option selected Trying to sort without any data
27	☒Fully implemented and working ☐Partially implemented ☐Not attempted	Menu Option 0	"Quitting the program"	"Quitting the program"	Exiting the program

28

Explain which testing strategy did you take (e.g., how you tested that the output is correct, different inputs, different values, wrong values, etc.)

1. Basic functionality test:
- I would first run the code to make sure it compiles and runs without any errors.
- Then, I would go through the menu and test each option to ensure that it functions as expected. For example, I would buy a ticket and check if the seat is marked as taken and if the ticket information is correctly stored. I would also cancel a ticket and check if the seat is marked as available again.
- I would verify that the available seats are correctly displayed and that the tickets are sorted by price as expected.

2. Boundary and limit cases tests:
- I would test the code with the minimum and maximum values for the row number, seat number, and price to ensure that the code handles these cases correctly.
- I would test the code with an empty file, a file with invalid data, and a file with valid data to ensure that the code can handle different types of input.

3. Error handling and validation tests:
- I would intentionally enter invalid input to test the code's error handling and validation. For example, I would enter a non-integer value when prompted for an input, enter an invalid row or seat number, or enter a negative price.

4. Load and Save functionality tests:
- I would test the load and save functions to ensure that the data is correctly read from and written to the file. I would verify that the data stored in the file is in the correct format and can be loaded back into the program without any errors.

5. Performance and scalability tests:
- I would test the code with a large number of tickets to ensure that it can handle a large amount of data without slowing down or crashing.
- I would test the code with multiple users accessing the system simultaneously to ensure that it can handle concurrent requests without any issues.

By testing the code with these different scenarios, I can ensure that the program is functioning as expected and is robust enough to handle different types of input and situations.

29

Did you include comments in your code? Is your code idented? Did you use your own functions? Are your variable names informative?
Comments are included in the program and for clear code I have used indentations and my own functions and also descriptive variable names

- Allows users to purchase tickets by specifying a row and seat number, and providing their personal information (name, surname, email).
- Allows users to cancel tickets by specifying the row and seat number.
- Allows users to view available seats in each row.
- Allows users to view the seating area (i.e., a visual representation of the rows and seats).
- Allows users to save all booking data to a file.
- Allows users to load booking data from a file.
- Allows users to view ticket information, including the total price of all tickets purchased.
- Allows users to sort tickets by price.

This uses arrays and ArrayLists to store and manipulate booking data, and includes basic error checking and input validation. The program also includes a simple color-coding scheme to improve readability of the output in the command-line interface.

Are there any parts of the coursework which you would like to get feedback?

DEMO: You will have to demonstrate your understanding of your code during a tutorial (week 10 or 11). Remember to reference any websites, or technologies that you used in this coursework. Tasks 9-15 will not be marked if you do not attend the demo.