CSD 1233 – Test 02

1. You have been hired by a company that owns hundreds of movie theatres across the country. The company wants you to build a system that allows its patrons to book reserve seats in its movie theaters at the time of ticket purchase.

   The system must support the following functionality:
   - Reserve seat(s) in the theatre (***reserveSeats***)
   - Cancel seat(s) reservation (***cancelSeats***)
   - Remove empty seats between patrons in a row (***removeEmptySeatsRow***)
   - Search for seats given booking number (***search***)
   - Display total number of seats reserved in whole theatre (***totalBooked***)

   Assume all theatres have 8 rows each with 10 seats for a total capacity of 80 seats.
   - NOTE: Use a 2D array to represent the seats
   - When initializing the array use the value 0 to represent empty seats

   IMPORTANT: Reservations are identified by a reservation number that begin a 1. Seats which are booked together share the same reservation number. Once a reservation number is used it cannot be reused even if the reservation is cancelled.
   - In Python, you may wish to use a global variable for the current reservation number although this is not a requirement
     - If you select **not to** use a global variable you will also have to provide the **currentBookingNumber** as an argument to the below functions

   Your program must have the following functions/procedures:

   ⇒ def **reserveSeats**(theatreArray, numberOfSeats, preference):
   - **PURPOSE**: Reserve **numberOfSeats** adjacent seats in any row in the theatre.
     - The method should find the first available adjacent seat(s) either from the first seat [0][0] or last seat [7][9] depending on above **preference**. If the seats cannot be reserved do not alter the **theatreArray**.
       - Note: Adjacent seats must be in the same row
   - Integer Array [8][10]: **theatreArray**
     - Array representing the current seat reservations for theatre
   - Integer: **numberOfSeats**
     - Number of adjacent seats to book
   - Integer: **preference**
     - A value of 1 represents a **preference** to sit at the back
     - A value of 2 represents a **preference** to sit at the front

- **Important:** If reservation is not possible return the **theatreArray** unmodified and display a message indicating the reservation could not be made.

⇒ def **reserveSeats**(theatreArray , numberOfSeats, row, columnStart):
- **PURPOSE**: Reserve **numberOfSeats** starting in **row** and starting at **columnStart**; if not enough adjacent seats are available in the row do not alter the **theatreArray**
- Integer Array [8][10]: **theatreArray**
  - Array representing the current seat reservations for theatre
- Integer: **numberOfSeats**
  - Number of adjacent seats to book
- Integer: **row**
  - Row number user wishes to book the seats in
  - **Note:** User will enter 1-8 (i.e. not 0 based starting value)
- Integer: **columnStart**
  - The first seat in the customer's booking
  - **Note:** User will enter 1-10 (i.e. not 0 based starting value)
- **Important:** If reservation is not possible return the **theatreArray** unmodified and display a message indicating the reservation could not be made.

⇒ def **cancelSeats**(theatreArray, bookingNumber):
- **PURPOSE**: Remove any seats with matching booking number. Remember empty seats should contain a value 0.
  - Remove all entries from **theatreArray** with matching **bookingNumber**.
- Integer Array [8][10]: **theatreArray**
  - Array representing the current seat reservations for theatre
- Integer: **bookingNumber**
  - The booking number associated with the seat(s)

⇒ def **removeEmptySeatsRow**(theatreArray,row):
- **PURPOSE**: Remove empty seats from the **row** specified; essentially, shifting every filled seat as far left as possible. Remember, empty seats should contain a value 0.
- Integer Array [8][10]: **theatreArray**
  - Array representing the current seat reservations for theatre
- Integer: **row**
  - Row number to remove empty seats from
  - Note: User will enter 1-8 (i.e. not 0 based starting value)

⇒ def **search**(theatreArray, bookingNumber):
- **PURPOSE**: Find all the seats in the **theatreArray** with matching **bookingNumber**. This function does not return the values but prints them out in the format similar to: "Row: 1, Seat 10"
- Integer Array [8][10]: **theatreArray**
  - Array representing the current seat reservations for theatre
- Integer: **bookingNumber**
  - The booking number associated with the seat(s)
- **Important:** If no seats with booking number can be found print the message: "No seats with specified booking number could be located."

⇒ def **totalBooked**(theatreArray):
- **PURPOSE**: Return an integer representing the number of seats booked in the theatre.
- Integer Array [8][10]: **theatreArray**
  - Array representing the current seat reservations for theatre

⇒ def **displayMap**(theatreArray):
- **PURPOSE**: Print the theatre map (see example output)
- Integer Array [8][10]: **theatreArray**
  - Array representing the current seat reservations for theatre

Important Information:

- Customers when entering the theatre see row 1 at the front and row 8 at the back; however, in Python we would say Row index 7 is the front and row index 0 is the back. So users of your application will enter row 1 as front and your code must convert this to row index 7.

	Seat 1	Seat 2	Seat 3	Seat 4	Seat 5	Seat 6	Seat 7	Seat 8	Seat 9	Seat 10
Row 8										
Row 7										
Row 6										
Row 5										
Row 4										
Row 3										
Row 2										
Row 1										

FRONT OF THEATRE

- Validate in the main that users are entering a valid menu option (1-8), row (1-8) and seat number (1-10).
-

Sample Output:

MENU

1. Reserve seats

2. Reserve seats with specific starting row and column
3. Cancel reservation
4. Remove empty seats from specific row
5. Search for reservation
6. Total seats booked
7. Display theatre map
8. Exit application

****

User selection: 7

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

MENU

1. Reserve seats
2. Reserve seats with specific starting row and column
3. Cancel reservation
4. Remove empty seats from specific row
5. Search for reservation
6. Total seats booked
7. Display theatre map
8. Exit application

****

User selection: 1

Enter number of adjacent seats do you require: 3
Preference of seating (1 for back, 2 for front): 2
MENU

1. Reserve seats
2. Reserve seats with specific starting row and column
3. Cancel reservation
4. Remove empty seats from specific row
5. Search for reservation
6. Total seats booked
7. Display theatre map
8. Exit application

****

User selection: 7

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	1

MENU

1. Reserve seats
2. Reserve seats with specific starting row and column
3. Cancel reservation
4. Remove empty seats from specific row
5. Search for reservation
6. Total seats booked
7. Display theatre map
8. Exit application

****

User selection: 2

Enter number of adjacent seats do you require: 3
Enter row number you wish to book seats: 1
Enter column/seat number you want your seats to begin at: 3

MENU

1. Reserve seats
2. Reserve seats with specific starting row and column
3. Cancel reservation
4. Remove empty seats from specific row
5. Search for reservation
6. Total seats booked
7. Display theatre map
8. Exit application

****

User selection: 7

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	2	2	2	0	0	1	1	1

MENU

1. Reserve seats
2. Reserve seats with specific starting row and column
3. Cancel reservation
4. Remove empty seats from specific row
5. Search for reservation
6. Total seats booked
7. Display theatre map
8. Exit application

****

User selection: 4

Which row would you like to remove empty seats from: 1

MENU

1. Reserve seats
2. Reserve seats with specific starting row and column
3. Cancel reservation
4. Remove empty seats from specific row
5. Search for reservation
6. Total seats booked
7. Display theatre map
8. Exit application

****

User selection: 7

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
2	2	2	1	1	1	0	0	0	0

MENU

1. Reserve seats
2. Reserve seats with specific starting row and column

3. Cancel reservation
4. Remove empty seats from specific row
5. Search for reservation
6. Total seats booked
7. Display theatre map
8. Exit application

****

User selection: 6

The total number of seats reserved in the theatre is 6.

MENU

1. Reserve seats
2. Reserve seats with specific starting row and column
3. Cancel reservation
4. Remove empty seats from specific row
5. Search for reservation
6. Total seats booked
7. Display theatre map
8. Exit application

****

User selection: 5

Reservation number are you searching for: 1

SEATS INCLUDED:
Row: 1, Seat 4
Row: 1, Seat 5
Row: 1, Seat 6

MENU

1. Reserve seats
2. Reserve seats with specific starting row and column
3. Cancel reservation
4. Remove empty seats from specific row
5. Search for reservation
6. Total seats booked
7. Display theatre map
8. Exit application

****

User selection: 3

Reservation number you would like to cancel: 2

MENU

1. Reserve seats
2. Reserve seats with specific starting row and column
3. Cancel reservation
4. Remove empty seats from specific row
5. Search for reservation
6. Total seats booked
7. Display theatre map
8. Exit application

****

User selection: 7

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	0

MENU

1. Reserve seats
2. Reserve seats with specific starting row and column
3. Cancel reservation
4. Remove empty seats from specific row
5. Search for reservation
6. Total seats booked
7. Display theatre map
8. Exit application

****

User selection: 1

Enter number of adjacent seats do you require: 5
Preference of seating (1 for back, 2 for front): 1

3	3	3	3	3	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

0	0	0	1	1	1	0	0	0	0

MENU

1. Reserve seats
2. Reserve seats with specific starting row and column
3. Cancel reservation
4. Remove empty seats from specific row
5. Search for reservation
6. Total seats booked
7. Display theatre map
8. Exit application

****

User selection: 8

Thanks for using the application!