React Teaser

Question:
In this project, you will write a program that can be used by management of Cinemax theater to assign and book seats, and sell tickets for performances.

The cinema has 16 rows, with 20 seats in each row. The seats at the mid-column (number 5 to 15) of the last 2 rows closest to the doors (row A and B) are Twin seats, which will be sold in pair, side by side (e.g., buy A5 means paying for both A5 and A6). The rest of the seats in the row A and B, and all seats in row C to F are Very VIP (VVIP) seats. Row G to L are VIP seats, and row M to Z are Economy seats. The map of the cinema is shown in Figure 1. The prices of the seats are shown in Table 1.

Table 1: Price for different seat category.

Seat	**Price (in shillings)**
Twin Seats	25,000
Very VIP	100,000
VIP	50,000
Economy	20,000

Step 1: The program should display a screen that shows which seats are available and which are taken. Seats that are taken should be represented by a # symbol, seats that are available should be represented by a * symbol and o indicates that the seat is reserved (without payment). The first thing your program should do is initialize all of the seats to available (*) and display the seating chart.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
B	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
C	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
D	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
E	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
F	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
G	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
H	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
I	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
J	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
K	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
L	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
M	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
N	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
O	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
P	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

S C R E E N

Number of seats available: 320
Total sales: shs. 0

Step 2: Each row in the cinema has a different ticket price as shown in table 1. Your program should read the ticket price of each row from an input file called prices.txt. Your program should have variables tracking the total number of tickets sold and the total revenue for all tickets sold.

Step 3: Your program should allow the user to sell tickets one at a time. The user should be able to sell as many tickets as they would like. Do this with some sort of prompt or menu asking the user if they would like to sell another ticket. Don't forget to validate input data if you need to: When tickets are being sold, do not accept row or seat numbers that do not exist. When someone requests a particular seat, the program should make sure that seat is available before it is sold.

To allow the user to sell a ticket your program should have the user enter a row number and a seat number for the ticket they would like to sell. The program should do four things with this information:

1. It should check to see if the seat is available. If the seat is taken the program should not allow the user to sell the ticket. If this happens, print a message to the user saying the ticket is not available and prompt the user to see if they would like to sell another ticket.
2. If the seat is available the program should update the seating chart by putting a taken symbol (#) in that seat's position in the chart.
3. The program should then look up the row price for the seat sold. Your program should have a variable tracking the total revenue, the price of the seat sold should be added to this total after each sale.
4. Your program should have a variable tracking the total tickets sold. The next thing your program should do when selling a ticket is update the total tickets sold.

Step 4: Once the user is finished selling tickets print out an updated seating chart followed by the total tickets sold and the total revenue generate from those tickets.

NOTE: You are required to use two arrays in this program, one for the seating chart and one to store the prices for each row. You are also required to use two functions: one to display the seating chart and one to read in the price per row data and store it in the array with the prices for each row in it. You may use other functions if you want to.

Furthermore, your program prompts the operator to enter the following information:
i. Number of seats required
ii. Desired seat category (Twin, VVIP, VIP or Economy)

Upon entering the information, the program is to list the available seats matched with the requirements. The list contains the available seats located side by side, in the same row and column that will meet the number of seats required. The list is shown with the starting seat number and the number of seats available after the seat. The operator can then choose to update the status of the seat. For example,

Enter number of seats required: 8
Desired seat category: VVIP
Available seats:
C5, 100000
D5, 100000
E5, 100000
F5, 100000

Choose seat: C7
Action: x

Your program then outputs the seating plan to the operator with updated information, as shown in the following.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

B	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
C	*	*	*	*	*	*	#	#	#	#	#	#	#	#	*	*	*	*	*	*
D	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
E	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
F	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
G	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
H	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
I	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
J	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
K	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
L	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
M	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
N	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
O	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
P	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

S C R E E N

Number of seats available: 312
Total sales: shs. 800,000

Enter number of seats required: 13
Desired seat category: VIP

Available seats: None
Please try again!

If there is no available seat in a row to match the required number of seats, the operator should break down the number to 2 smaller numbers so that customers are to be seated in separate rows. A ticket buyer should be allowed to swap seats with another.

The above described only the functions of your program under the Seat Assignment menu. The complete program should be menu-driven with the following choices:
i. Seat Assignment
ii. Payments (for reserved seat)
iii. Reset Seating Plan
iv. Exit

With the Payment menu available, your program will allow the operator to change the reserved seats to available seats, considering all reserved seats will expired eventually if payment is not received after some time.

GOOD LUCK!!