Theater Seating (Modified from book, only use vectors)

Write a program that can be used by a small theater to sell tickets for performances.

1. The DEFAULT size of theater's auditorium is 15 rows of seats, with 30 seats in each row. Default seat price is $10.50
   1.1. Seats will be referenced as R<#>S<#> *for example: R10S12 (row 10 seat 12)*
   1.2. The program will have a setup menu option which will ask the user to set the following ***(using this option resets/overwrites all previous settings, you must warn your user)***:
      - Number of rows
      - Number of seats per row
      - Default seat price
      - List of special row prices: entered as a spaces-separated list in one line as: R<#>$<#> *(example: R10$11.25 R7$15.00 R2$20) … these values will overwrite the default value for the given rows*
      - List of blocked seats listed entered in one line *(example R15S10 R7S11 R2S3)*
   1.3. The program will auto-save these settings in a "setup.txt" file
   1.4. If the "setup.txt" file was not found at start time, the theatre seating will default to 15rows x 30 seats. HOWEVER, these values will change whenever the user uses the setup submenu above. **The change will happen by resizing the vectors to new size (use proper functions)**
   1.5.
2. The program should display a screen that shows which seats are available and which are taken. For example, the following screen shows a chart depicting each seat in the theater. Seats that are taken are represented by an * symbol, blocked seats are represented by **X** symbol, and seats that are available are represented by a # symbol:

```
               Seats
      123456789012345678901234567890
Row 1 ***###***###*########*****####
Row 2 ####*************####*******##
Row 3 **###**********########****###
Row 4 **######**************##******
Row 5 ********#####*********########
Row 6 ##############************####
Row 7 #######************########XXX
Row 8 ************##****############
Row 9 #########*****############****
Row 10#####*************###X########
Row 11#**********#################**
Row 12#####XXX#####********########*
Row 13###***********########**######
Row 14##############################
Row 15##############################
```

Here is a list of tasks this program must perform:

3. The program should display a seating chart similar to the one shown above. The user may enter the row and seat numbers for tickets being sold. Every time a ticket or group of tickets is purchased, the program should display the total ticket prices and update the seating chart.
4. The program should keep a total of all ticket sales. The user should be given an option of viewing this amount.
   4.1. The program should also give the user an option to see a list of how many seats have been sold, how many seats are available in each row, and how many seats are available in the entire auditorium.

BE CREATIVE and think of it as a real-life use.

5. Purchases:
   5.1. Users may request buying x-number of adjacent seats, the program will auto-find the seats
   5.2. Users may request buying x-number of adjacent seats at a given row, the program will auto-find the seats starting at this row and moving towards the back (higher row numbers) till something is found of error indication it was not able to locate such a set.
   5.3. Users may request a specific seat
6. There must be an option to clear all sales
7. There must be an exit option. All sales must be saved at exit time and loaded at startup time

Input Validation: 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.

Hints:

- Although this program requires a 2D vector, it should not be a vector in a vector (i.e. vector<vector<char>> …. Think of a better data representation
- This program must be separated into proper functions or it will become too difficult and complicated
-