CSAP - ArrayList Lab Seating Chart

In this assignment you will use a 2 dimensional array to create a seating chart from an ArrayList of Student objects.

The `Student` class

Instance Variables

- Student name - `String`
- number of absences - `int`

Methods

- `public Student (String n, int abs)` - the constructor
- `public String getName()` - returns student name
- `public int getAbsenceCount()` - returns the student's absence
- `public String toString()` - displays the student name and number of absences

The `SeatingChart` class

Instance Variables

- `seats` - a 2 dimensional array of Student object references

Methods

- `public SeatingChart (List <Student> studentList, int rows, int cols)` - the constructor to instantiates the two dimensional array `seats` and populates it from the ArrayList `studentList` in column by column order.

  For example, suppose the `List<Student> Juniors17` contains references to `Student` objects in the following order:

“Anna” 3	“Ben” 1	“Caren” 4	“David” 1	“Evan” 5	“Fran” 9	“Gina” 2	“Holly” 6	“Irene” 1	“Justin” 3

  A `SeatingChart` object created with `new SeatingChart (Juniors17, 3, 4)` would have seats initialized with the following values:

	0	1	2	3
0	“Anna” 3	“David” 1	“Gina” 2	“Justin” 3
1	“Ben” 1	“Evan” 5	“Holly” 6	`null`
2	“Caren” 4	“Fran” 9	“Irene” 1	`null`

- `public int removeAbsentStudents (int allowedAbs)` - removes all students with more than `allowedAbs` absences from the seating chart, replacing the array with `null` and returns the number of students that were removed.

  For example, suppose `room213`, a `SeatingChart` object has been created and contains the following students and their absences.

	0	1	2	3
0	"Anna" 3	"David" 1	"Gina" 2	"Justin" 3
1	"Ben" 1	"Evan" 5	"Holly" 6	null
2	"Caren" 4	"Fran" 9	"Irene" 1	null

  After the call `room213.removeAbsentStudents(4)` has been executed, the array `seats` would contain the following values and return `3`.

	0	1	2	3
0	"Anna" 3	"David" 1	"Gina" 2	"Justin" 3
1	"Ben" 1	null	null	null
2	"Caren" 4	null	"Irene" 1	null

- `public boolean switchSeats (int RowA, int ColA, int RowZ, int ColZ)` - switches the student sitting in `RowA` and `ColA` with the student sitting in `RowZ` and `ColZ`. You may assume `RowA, ColA, RowZ` and `ColZ` are valid row and column numbers. Return `true` if both seats have student objects or one seat has a null. Return `false` if both seats have `null`.

  Example 1: Suppose `room213` has been created and contains the following students in the following seats.

	0	1	2	3
0	"Anna" 3	"David" 1	"Gina" 2	"Justin" 3
1	"Ben" 1	null	null	null
2	"Caren" 4	null	"Irene" 1	null

  After the call `room213.switchSeats (2,0,0,3) and` has been executed, the array `seats` would contain the following values and return `true`.

	0	1	2	3
0	"Anna" 3	"David" 1	"Gina" 2	"Caren" 4
1	"Ben" 1	null	null	null
2	"Justin" 3	null	"Irene" 1	null

Example 2: Suppose `room213`, a `SeatingChart` object contains the following students in the following seats.

	0	1	2	3
0	“Anna” 3	“David” 1	“Gina” 2	“Justin” 3
1	“Ben” 1	null	null	null
2	“Caren” 4	null	“Irene” 1	null

After the call `room213.switchSeats (1,3,2,2) and` has been executed, the array `seats` would contain the following values and return `true`.

	0	1	2	3
0	“Anna” 3	“David” 1	“Gina” 2	“Justin” 3
1	“Ben” 1	null	null	“Irene” 1
2	“Caren” 4	null	null	null

Example 3: Suppose `room213`, a `SeatingChart` object contains the following students in the following seats.

	0	1	2	3
0	“Anna” 3	“David” 1	“Gina” 2	“Justin” 3
1	“Ben” 1	null	null	“Irene” 1
2	“Caren” 4	null	null	null

After the call `room213.switchSeats (1,1,2,2)` has been executed, the array `seats` would remain unchanged and the method would return `false`.

- `public boolean isValidRow (int r)` - returns true if r is a valid row number, false otherwise
- `public boolean isValidCol (int r)`- returns true if r is a valid col number, false otherwise
- `public String toString()` - returns the contents of each `Student` object and the row and column.

```
Anna, 3, 0, 0
David, 1, 0, 1
Gina, 2, 0, 2
Justin, 3, 0, 3
Ben, 1, 1, 0
null, 1, 1
null 1, 2
Irene, 1, 1, 3
Caren, 4, 2, 0
null, 2, 1
null, 2, 2
null, 2, 3
```