Design/Coding Exercise

- You may not use any external libraries to solve the problem itself, but you may use external libraries or tools for building or testing purposes.

How we evaluate your code

We will be looking at a number of things including the design aspect of your solution and your object oriented programming skills. Whilst these are small problems, we expect you to submit what you believe is “production-quality” code that you would be able to run, maintain and evolve. You don’t need to “gold plate” your solution, but we are looking for something more than a bare-bones algorithm. You should submit code that you would be happy to produce in a real project, or that you would be happy to receive from a colleague. We recommend you to provide adequate tests to indicate full code coverage. We also recommend you follow good code hygiene and clean code practices to reduce cyclomatic complexity of your classes.

How to submit your code

Please upload the code to github, and submit a link.

Problem: Theatre Seating

You run a small theater and each month, you have patrons mail in requests for pre-sale tickets. You need to process these ticket requests and either tell them where their party will sit or explain to the patron why you can't complete their order.

You have a few rules that you need to follow when you fill the orders

1. Fill as many orders as possible
2. Put parties as close to the front as possible.
3. If there are not enough seats available in the theater to handle a party, tell them "Sorry, we can't handle your party."
4. Each party must sit in a single row in a single section. If they won't fit, tell them "Call to split party".

Your program must parse a theater layout and a list of ticket requests and produce a list of tickets or explanations in the same order as the requests.

The theater layout is made up of 1 or more rows. Each row is made up of 1 or more sections separated by a space.

After the theater layout, there is one empty line, followed by 1 or more theater requests. The theater request is made up of a name followed by a space and the number of requested tickets.

Sample input:

```
6 6
3 5 5 3
4 6 6 4
2 8 8 2
6 6

Smith 2
Jones 5
Davis 6
Wilson 100
Johnson 3
Williams 4
Brown 8
Miller 12
```

Your program must produce results to standard output in the same order as the requests, with the name of the person who requested the ticket and either the row and section of the ticket or the explanations "Sorry, we can't handle your party" or "Call to split party."

Sample output:

```
Smith Row 1 Section 1
Jones Row 2 Section 2
Davis Row 1 Section 2
Wilson Sorry, we can't handle your party.
Johnson Row 2 Section 1
Williams Row 1 Section 1
Brown Row 4  Section 2
Miller Call to split party.
```