Multi Dimensional Array Evaluation Task

Suppose your ST is designing a new seating arrangement for quiz exams for your section. Usually, seating arrangements are made based on student ID. In this way, students always know who is going to sit beside who. So there is a high chance of copying from others and you don't want that. Need to maintain the Zero Tolerance Plagiarism Policy! That's why, this time, you want to give them a surprise by making a totally new arrangement for the students so that nobody knows who is going to sit beside who in the exam. Instead of arranging the seats based on Student ID, you told your ST to use something new this time which is the previous quiz marks of all the students. The seats are arranged in such a way that it can be imagined as a square matrix where each cell of that matrix is the last quiz mark of a student. Your goal is to make sure that the difference between the summation of all last quiz marks of the k-th row and k-th column of that classroom is equal **[(sum(row1)-sum(col1))=(sum(row2)-sum(col2))=...=(sum(rowN)-sum(colN))].**

Now, your ST has designed the arrangement and submitted it to you. Your task will be to check whether his/her arrangement fulfills the requirements or not. As a CSE graduate, you can write a Python script to check the validity of the arrangement.
The seating arrangement is done in such a way where the row and column number is the same (m x m dimension)

[Note: If you are wondering about the negative marks of the previous quiz, then you can consider that he/she was peeking from a peer's quiz copy. So we did what we had to do :( ]

Sample Input	Output
| 3 | -9 | 7 | 0 | |---|---|---|---| | 3 | 7 | -1 | 6 | |---|---|---|---| | 4 | 19 | 2 | 0 |	The sum of the each row: [1 15 25 8] The sum of the each column: [10 24 5 10] Invalid

---------- | 0 | 7 | -3 | 4 | ----------	
| 1 | 2 | 7 | 0 | ---------- | 3 | 5 | 2 | 6 | ---------- | 4 | 2 | 0 | 0 | ---------- | 2 | 7 | -3 | 4 | ----------	The sum of the each row: [10 16 6 10] The sum of the each column: [10 16 6 10] Valid
| 3 | 2 | 7 | 0 | ---------- | 3 | 7 | 2 | 6 | ---------- | 4 | 2 | 2 | 0 | ---------- | 2 | 7 | -3 | 6 | ----------	The sum of the each row: [12 18 8 12] The sum of the each column: [12 18 8 12] valid