Problem 1: Magic squares. An $n \times n$ matrix that is filled with the numbers $1, 2, 3, \ldots, n^2$ is a magic square if the sum of the elements in each row, in each column, and in the two diagonals is the same value.

16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

Write a program that reads in 16 values from the keyboard and tests whether they form a magic square when put into a 4×4 vector. You need to test three features:

- (25 pt) Does each of the numbers $1, 2, ..., 16$ occur once (and only once) in the user input?
- (25 pt) Can you process the input into the form of 2 dimensional vector form?
- (30 pt) When the numbers are put into a square, are the sums of the rows, columns, and diagonals equal to each other?

1. You should use the given template. Do not include other header files. For the algorithm header file, you may only use the sort function if needed.
2. Do not modify the code below the line in the template. Do not forget to declare the ownership in the beginning of `MagicSquares.cpp`.
3. For simplicity, we suppose the user input integers are always positive integers. You don't need to worry about other cases.

Here are examples of sample runs (the second line is user input)

```
Please input a sequence of postive integers, ending with Q:
16 3 2 13 5 10 11 8 9 6 7 12 4 15 14 1 Q
This is a magic square.
```

The above example forms a magic square as shown in the picture.
or

```
Please input a sequence of postive integers, ending with Q:
2 2 3 5 Q
invalid input!
```

or

```
Please input a sequence of postive integers, ending with Q:
16 3 2 13 5 10 11 8 9 6 7 12 4 15 1 14 Q
This is not a magic square.
```

Instructions:

- All code must be written originally by yourself. You are not allowed to (even partially) copy code from anyone else. Incident of cheating or plagiarism will be reported to the Dean's office and results in a zero grade in this assignment.
- (2.5pt) Fill in the template `MagicSquares.cpp` without changing the file name, and only submit it to Gradescope.
- (5pt) Declare the ownership in the beginning of your file. See details in HW1.
- (80pt) Implement all functions correctly.
- (12.5pt) Write your code with good coding practices, including commenting your code, using descriptive variable names, using constant variables, etc.
- Code compiles with Visual Studio 2022 and solves the questions. Students may lose the majority of points if their code doesn't compile with VS 2022. To receive full credits, the output must look EXACTLY the same as instructed above, including words, spaces, symbols, etc. Your code should not only work for the above examples, but also work for other different inputs.