Vertical Additive Symmetric Matrix

Matrixes

A **Matrix** is a 2-dimensional array represented as a row x column data structure.

Vertical Additive Symmetric Matrix

A matrix of integer numbers is defined to exhibit vertical additive symmetry if the sum of the numbers in the columns of the matrix exhibits vertical symmetry … if the total sum for the columns is the same. In a list with an odd number of columns, the middle column is considered symmetric with itself.

Examples:

Vertical Additive Symmetric Matrixes

```
     |
 2   56    3        1     7  |   11     1
 7   12    8        2    10  |   10     1
 3    0    1        3     3  |   -1     4
--   --   --       --    --  |   --    --
12    |   12        6    20  |   20     6
|---------|             |-------|
                    |------------------|
```

Not Vertical Additive Symmetric Matrixes

```
     |
 3   56    3        2     7  |   11     1
 7   12    8        2    10  |   10     1
 3    0    1        3     3  |   -1     4
--   --   --       --    --  |   --    --
13    |   12        7    20  |   20     6
|---- X----|             |---|---|
                    |--------X---------|
```

Your program must read a file of candidate matrixes that are specified in the file the following way:

- The first file row contains 2 space delimited numbers for the number of rows and columns in the matrix that follows.
- The matrix that follows consists of numbers in the form of multiple lines of space delimited numbers. Each number can be at most 4 digits long.
- In this project, no input candidate matrix should have more than 20 rows and columns.
- The file name to be read is to be **matrixes.txt**.
- You must create a a matixes.txt file that must contain at least 10 candidate matrixes. 6 of the matrixes must be vertical additive symmetry matrixes and 4 matrixes must not have vertical additive symmetry.

You are only responsible for checking the following regarding the input file:

- If the file open fails
- Correctly detecting the end of the file
  - The program should operate correctly even if there are blanks at the end of the file.
- The program does not have to check for data or format errors in the file.

The submittal must include a **matrixes.txt** file that demonstrates the criteria mentioned in the above section of the **matixes.txt** required content.

The program must:

- Process each matrix one at a time
- Matrix displays can assume that each matrix entry is made up at most 4 integer places.
- All data displays must be in a in a well formatted tabular style using spaces
- Display the inputted matrix
- Display a data list of the column vertical additive sums
- Display a message that states whether this matrix has vertical additive symmetry or not
- Display the sorted matrix rows
  - Display each row of the matrix in ascending order, one row per line of output
- Display a message that pauses the screen and states to enter a key to continue

Review the rules in the Required Best Practices document posted to the eLearning system. Ensure your program is well documented, well structured, modular and uses meaningful variable names.

Vertical Additive Symmetric Matrix

Use functions, with parameters, as necessary for good structure and readability. Do not use global variables.

All functions must exist after the main () functions, so use function prototypes.

Use the following code to pause the screen:

```
cout << "Press the enter key once or twice to continue..." << endl; cin.ignore(); cin.get();
```

There is a system dependency that may require hitting the key twice for correct behavior.

See the sample run output below to guide your programming and display outputs.

Example Run:

Given the following in the input file:

```
2 3
12 25 45
56 98 34
3 4
2 1 7  7
4 6 1 -2
1 1 0  2
```

Output:

```
Input:
          12          25          45
          56          98          23
Sums :
          68         123          68

Vertical additive symmetry : Yes

Sorted:
          12          25          45
          23          56          98

Press the enter key once or twice to continue...


Input:
           1           1           7           7
           4           6           1          -2
           1           1           0           2
Sums :
           6           8           8           7

Vertical additive symmetry : No

Sorted:
           1           1           7           7
          -2           1           4           6
           0           1           1           2

Press the enter key once or twice to continue...


Program Done
```