

8.1（求矩阵中各列数字的和）

编写一个方法，求整数矩阵中特定列的所有元素之和，使用下面的方法头：

```
public static double sumColumn(double[][] m, int columnIndex)
```

编写一个测试程序，读取一个 3×4 的矩阵，然后显示每列元素的和。

下面是一个运行示例。

```
Enter a 3-by-4 matrix row by row:  
1.5 2 3 4 ↵ Enter  
5.5 6 7 8 ↵ Enter  
9.5 1 3 1 ↵ Enter  
Sum of the elements at column 0 is 16.5  
Sum of the elements at column 1 is 9.0  
Sum of the elements at column 2 is 13.0  
Sum of the elements at column 3 is 13.0
```

8.13 （定位最大的元素）

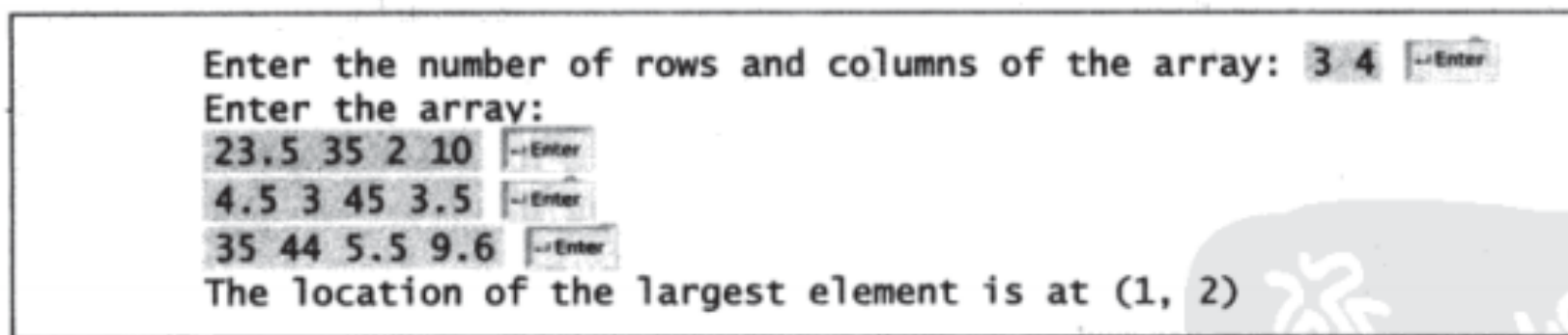
编写下面的方法，返回二维数组中最大元素的位置。

```
public static int[] locateLargest(double[][] a)
```

返回值是包含两个元素的一维数组。这两个元素表示二维数组中最大元素的行下标和列下标。

编写一个测试程序，提示用户输入一个二维数组，然后显示这个数组中最大元素的位置。

下面是一个运行示例。



8.19（模式识别：联系的四个相等的数）

编写下面的方法，测试一个二维数组是否有四个连续的数字具有相同的值，这四个数可以是水平方向的、垂直方向的或者对角线方向的。

```
public static boolean isConsecutiveFour(int[][] values)
```

编写一个测试程序，提示用户输入一个二维数组的行数、列数以及数组中的值。如果这个数组有四个连续的数字具有相同的值，就显示**true**；否则，显示**false**。

下面是结果为**true**的一些例子。

0	1	0	3	1	6	1
0	1	6	8	6	0	1
5	6	2	1	8	2	9
6	5	6	1	1	9	1
1	3	6	1	4	0	7
3	3	3	3	4	0	7

0	1	0	3	1	6	1
0	1	6	8	6	0	1
5	5	2	1	8	2	9
6	5	6	1	1	9	1
1	5	6	1	4	0	7
3	5	3	3	4	0	7

0	1	0	3	1	6	1
0	1	6	8	6	0	1
5	6	2	1	6	2	9
6	5	6	6	1	9	1
1	3	6	1	4	0	7
3	6	3	3	4	0	7

0	1	0	3	1	6	1
0	1	6	8	6	0	1
9	6	2	1	8	2	9
6	9	6	1	1	9	1
1	3	9	1	4	0	7
3	3	3	9	4	0	7

8.25（马尔科夫矩阵）

一个 $n \times n$ 的矩阵被称为正马尔科夫矩阵，当且仅当每个元素都是正数，并且每列的元素之和为1。编写下面的方法来检测一个矩阵是否是一个马尔科夫矩阵。

```
public static boolean isMarkovMatrix(double[][] m)
```

编写一个测试程序，提示用户输入一个 3×3 的double值的矩阵，测试它是否是一个马尔科夫矩阵。

下面是一个运行示例。

```
Enter a 3-by-3 matrix row by row:
```

```
0.15 0.875 0.375 --Enter
```

```
0.55 0.005 0.225 --Enter
```

```
0.30 0.12 0.4 --Enter
```

```
It is a Markov matrix
```

```
Enter a 3-by-3 matrix row by row:
```

```
0.95 -0.875 0.375 --Enter
```

```
0.65 0.005 0.225 --Enter
```

```
0.30 0.22 -0.4 --Enter
```

```
It is not a Markov matrix
```

8.27（列排序）

使用下面的方法实现二维数组的列排序。返回一个新的数组，并且原数组保持不变。

```
public static double[][] sortColumns(double[][] m)
```

编写一个测试程序，提示用户输入一个 3×3 的double值的矩阵，显示一个新的每列排好序的矩阵。

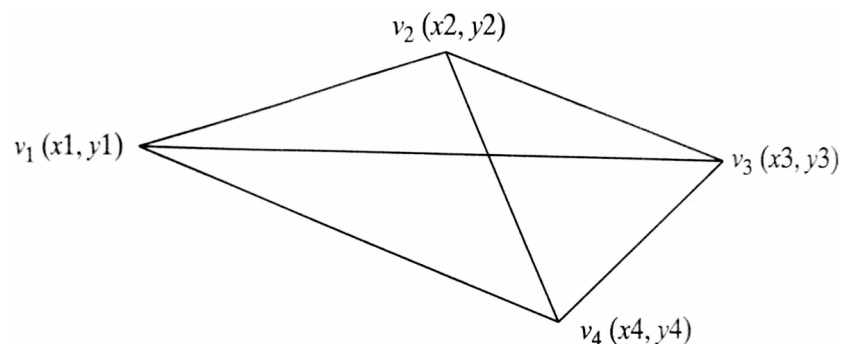
下面是一个运行示例。

```
Enter a 3-by-3 matrix row by row:
0.15 0.875 0.375 ↵ Enter
0.55 0.005 0.225 ↵ Enter
0.30 0.12 0.4 ↵ Enter

The column-sorted array is
0.15 0.0050 0.225
0.3 0.12 0.375
0.55 0.875 0.4
```

8.33（几何：多边形的子面积）

一个具有多个顶点的凸多边形被分为四个三角形，如下图所示。



编写一个程序，提示用户输入四个顶点的坐标，然后以升序显示四个三角形的面积。

下面是一个运行示例。

```
Enter x1, y1, x2, y2, x3, y3, x4, y4:
```

```
-2.5 2 4 4 3 -2 -2 -3.5
```

```
Enter
```

```
The areas are 6.17 7.96 8.08 10.42
```

8.35（最大块）

给定一个元素为0或者1的方阵，编写一个程序，找出一个元素都为1的最大的子方阵。程序提示用户输入矩阵的行数。然后显示最大的子方阵的第一个元素，以及该子方阵中的行数。

下面是一个运行示例。

```
Enter the number of rows in the matrix: 5 
```

```
Enter the matrix row by row:
```

```
1 0 1 0 1 
```

```
1 1 1 0 1 
```

```
1 0 1 1 1 
```

```
1 0 1 1 1 
```

```
1 0 1 1 1 
```

```
The maximum square submatrix is at (2, 2) with size 3
```