Old Mutual - CSSA
Computer Olympiad
DAY 1

Problems Overview

Problem	Willies Walks	Mine Sweeper	Block Town
Program name	WALK.EXE	MINE.EXE	BLOCK.EXE
Source name	WALK.PAS WALK.JAV WALK.CPP	MINE.PAS MINE.JAV MINE.CPP	BLOCK.PAS BLOCK.JAV BLOCK.CPP
Input file	WALK.IN	MINE.IN	BLOCK.IN
Output file	WALK.OUT	MINE.OUT	BLOCK.OUT
Time limit per test	10 seconds	10 seconds	10 seconds
Number of tests	10	10	10
Points per test	3	3	4
Total points	**30**	**30**	**40**

The maximum total score for Round I is 100 points.

Willie's Walks (30)

Description

The scene for this walk is an arrangement of cubes on a 4x4 square base as indicated below. (This should be a 3-D diagram, you'll just have to imagine the cubes rising out of the page.)

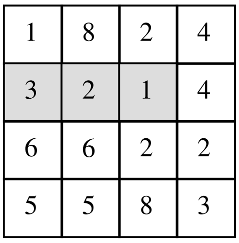
1	8	2	4
3	2	1	4
6	6	2	2
5	5	8	3

Each number indicates the number (i.e. height) of cubes at that location.

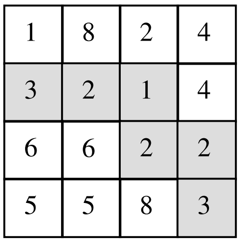
Willie likes to go for a walk on the cubes, but he can't step onto a cube that is more than one above or below the one he is on.

You must write a program to find the length of the longest walk he can make without stepping on any cube more than once.

There are two separate problems, in the first, Willie can only walk in a straight line, in the second, he can go North, South, East or West.



Part 1
(Longest straight walk)



Part 2
(Longest walk)

Input

The input will be four lines, each containing four integers representing the height h, $(1 \le h \le 8)$ of each stack of cubes.

Output

There will be two lines in the output, the first will be the length of the longest straight walk, the second will be the length of the longest walk.

Example 1

Input

```
1 8 2 4
3 2 1 4
6 6 2 2
5 5 8 3
```

Output

```
3
6
```

Example 2

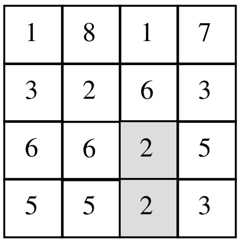
Input

```
1 8 1 7
3 2 6 3
6 6 2 5
5 5 2 3
```

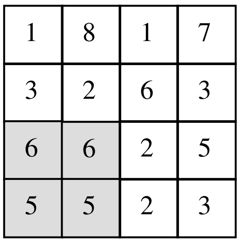
Output

```
2
4
```

The second example is shown below:



Part 1
(Longest straight walk)



Part 2
(Longest walk)

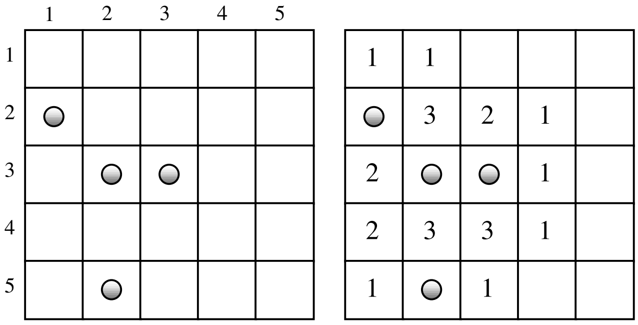
Note that there are a number of solutions to part 1. The walk for part 2 could start on any of the four shaded cubes.

Mine Sweeper (30)

Description

The well-known PC game, minesweeper is played on a square grid. Certain squares have mines placed on them, and other squares get a rating depending on the number of neighbouring mines. A player wants to know the ratings of the squares, and requests you to write a program to discover this.

The following minefield has four mines at (1, 2), (2, 3), (3, 3) and at (2, 5). The diagram on the right shows the ratings of the dangerous squares



	1	2	3	4	5
1					
2	●				
3		●	●		
4					
5		●			

1	1			
●	3	2	1	
2	●	●	1	
2	3	3	1	
1	●	1		

There are three squares with a rating of three and three with a rating of two, and seven with a rating of one.

Input

The first line contains N, (1 ≤ N ≤ 10) the length of the side of the square. This is followed by M, (1 ≤ M ≤ $N^2 – 1$), the number of mines, followed by M lines each containing the x and y co-ordinates of a mine.

Output

For each rating that occurs, you must output the rating and the number of squares with that rating. Output the most dangerous (i.e. highest) ratings first.

Example 1

Input

```
5
4
1 2
2 3
3 3
2 5
```

Output

```
3 3
2 3
1 7
```

Example 2

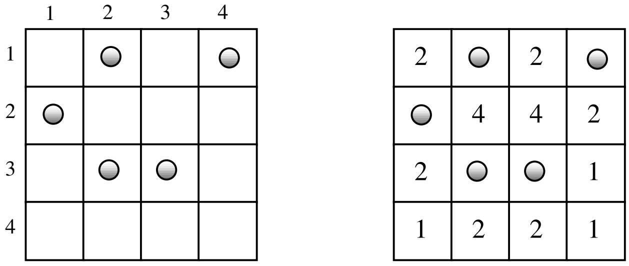
Input

```
4
5
1 2
2 1
2 3
3 3
4 1
```

Output

```
4 2
2 6
1 3
```

Here is the diagram for the second example, a 4x4 square with five mines.



	1	2	3	4
1		●		●
2	●			
3		●	●	
4				

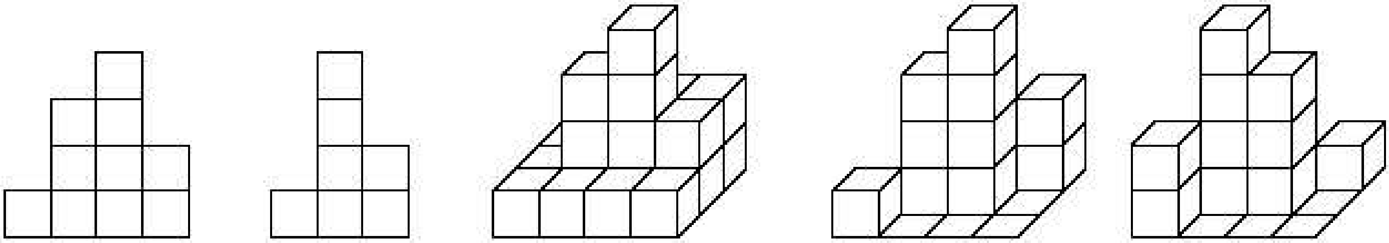
2	●	2	●
●	4	4	2
2	●	●	1
1	2	2	1

Note: the highest rating here is four, and there are two squares with that value, so the first line of the output is: “`4 2`”.

Block Town (40)

Description

Johnny likes playing with wooden blocks (cube bricks) usually he builds high towers, but today Johnny dreams of bigger plans. He wants to build a large town. His father has bought him a large table marked out in squares to match the blocks. Johnny decides to create a plan of such a town before he starts building it. He starts by marking out a rectangle on the table consisting of K x L squares. He wants to place the towers consisting of one or more blocks on some of the squares of the rectangle; the remaining squares will be empty. Because the rectangle is so large, Johnny is not going to plan exactly how many blocks he will put on every square. He only wants to decide about the front and right views of his town. He draws plans of the two views (two-dimensional projections) of the town. An example of these drawings and resulting towns is given below:



From the left: side view, front view, maximal town, minimal town (both front and back view)

Task

Johnny' s father is afraid they don' t have enough blocks to finish building Johnny' s planned town. You are required to write a program to compute the minimal and maximal amount of blocks with which a town corresponding to Johnny' s plans can be built. Moreover the program should decide whether it is possible to build a town satisfying the views.

Input:

The first line of input file TOWN.IN contains two positive integers K and L, the width and the length of the rectangle (expressed as numbers of blocks). Neither the width nor the length of the rectangle is greater than 100 000 bricks. The following lines of the input file contain the description of the front view of the town. The description consists of a series of heights (expressed as numbers of blocks) of visible buildings on each square from the left to the right. There is only one number on each line, i.e. the number of the lines with the front view description of the town equals K, the width of the rectangle. Similarly the next L lines of the input file contain the right view of the town. The heights of the wooden block towers are now specified from the front line to the back line. There is no building in the town with height exceeding 5000 blocks. The maximal number of blocks needed for building the entire town does not exceed 2 000 000 000.

Output:

Output file TOWN.OUT contains only one line. If it is not possible to build a town with the front and right sight views given, the text 'No solution.' is written out. In the other case two numbers will be written on the line and separated by a single space. The first one is the minimal and the second one is the maximal number of blocks Johnny needs to build the town to suit his plan.

Example 1:

TOWN.IN

```
4 3
1
3
4
2
1
4
2
```

TOWN.OUT

```
10 21
```

Example 2:

TOWN.IN

```
2 2
4
1
1
3
```

TOWN.OUT

```
No solution.
```