|  |  |
| --- | --- |
| **Little Elephant and Magic**  Problem code: LEMAGIC | * [**SUBMIT**](http://www.codechef.com/problems/LEMAGIC) * [**ALL SUBMISSIONS**](http://www.codechef.com/status/LEMAGIC) |

**All submissions for this problem are available.**

The Little Elephant from the Zoo of Lviv believes in magic.

He has a magic board **A** that consists of **N** rows and **M** columns. Each cell of the board contains an integer from **0** to **9** inclusive. The cell at the intersection of the **i**-th row and the **j**-th column is denoted as **(i; j)**, where **1** ≤ **i** ≤ **N** and **1** ≤ **j** ≤ **M**. The number in the cell **(i; j)** is denoted as **A[i][j]**.

The Little Elephant owns the only magic operation which can be described as follows. He chooses some integer **P** and some row or column, after that for each cell in the chosen row or column he adds number **P**to the number in this cell and take the result modulo **10** in order to keep numbers in the range **{0, 1, ..., 9}**. Our Little Magician wants to perform series of such operations to achieve some board for which certain characteristic called *level of the board* is maximal possible.

So what is the level of the board? Bluntly speaking it is the length of the longest non-increasing subsequence of cells of the board. Formally, the level of the board is the maximal integer **K** such that there exists such sequence of **different** cells **(i1; j1), (i2; j2), ..., (iK; jK)** for which

**1** ≤ **i1** ≤ ... ≤ **iK** ≤ **N**,   
**1** ≤ **j1** ≤ ... ≤ **jK** ≤ **M**,

and

**A[i1][j1]** ≥ **A[i2][j2]** ≥ ... ≥ **A[iK][jK]**.

Though, the magic operation, the Little Elephant owns, is quite powerful, there are some restrictions dictated by the Association of Cursed Magicians (ACM):

* The number **P** should be chosen in advance and should be the same for all operations.
* For each row the magic operation can be applied at most once.
* The same, of course, is true for columns.

Without these stupid restrictions of this stupid Association our hero could always achieve the maximal possible level **M + N − 1**. But now he is confused and asks you for help. Find the maximal level of the board **A** after making arbitrary number of magic operations according to the restrictions of ACM.

**Input**

The first line of the input contains a single integer **T**, the number of test cases. Then **T** test cases follow. The first line of each test case contains two space separated integers **N** and **M**, the sizes of the board. Each of the following **N** lines contains **M** one-digit numbers without spaces. The **i**-th line among them contains the numbers **A[i][1]**, ..., **A[i][M]**.

**Output**

For each test case output a single line containing a single integer, the maximal possible level of the board that Little Elephant can achieve under the restrictions of ACM.

**Constraints**

**1** ≤ **T** ≤ **10**  
**1** ≤ **N** ≤ **100**  
**1** ≤ **M** ≤ **100**  
**0** ≤ **A[i][j]** ≤ **9**

**Example**

**Input:**

2

2 2

11

10

3 4

3478

4268

7173

**Output:**

3

5

**Explanation**

**Case 1.** The board already has a sequence of **3** cells that satisfies all required constraints (without applying any operation). For example, one can choose, the sequence **(1; 1), (1; 2), (2; 2)**. It is also shown in the figure below (chosen cells are made bold):

**11**

1**0**

Let's formally validate this sequence of cells. Inequality **1** ≤ **i1** ≤ ... ≤ **iK** ≤ **N** takes the form **1** ≤ **1** ≤ **2**. Inequality **1** ≤ **j1** ≤ ... ≤ **jK** ≤ **M** takes the form **1** ≤ **2** ≤ **2**. Finally, inequality **A[i1][j1]** ≥ **A[i2][j2]** ≥ ... ≥ **A[iK][jK]**takes the form **1** ≥ **1** ≥ **0**. So all of them is satisfied, which means that the level of this board is at least **3**. But clearly, we can't have the required sequence of cells of length more than **3**. So **3** is the actual level of this board.

**Case 2.** The desired sequence of length **5** can be achieved by several values of **P**. Consider, for example,**P = 3**. At first let's apply the magic operation to the 1st row. We get the following transformation:

3478 → **6701**

4268 → 4268

7173 → 7173

Now let's transform the 1st column by the magic operation. We get:

6701 → **9**701

4268 → **7**268

7173 → **0**173

Finally we modify the 2nd column:

9701 → 9**0**01

7268 → 7**5**68

0173 → 0**4**73

Now we can take the following sequence of 5 cells to satisfy all needed constraints: **(1; 1), (2; 1), (2; 2), (3; 2), (3; 4)** (see the figure below):

**9**001

**75**68

0**4**7**3**

Just to reiterate we note that the inequality **A[i1][j1]** ≥ **A[i2][j2]** ≥ ... ≥ **A[iK][jK]** takes the form **9 ≥ 7 ≥ 5 ≥ 4 ≥ 3** for this sequence. One can check (for example, by brute force), that sequences of length more than **5**can't be achieved. So **5** is the answer.

|  |  |
| --- | --- |
| Author: | [witua](http://www.codechef.com/users/witua) |
| Tags | [cook28](http://www.codechef.com/tags/problems/cook28)[dp](http://www.codechef.com/tags/problems/dp)[dynamic-prog](http://www.codechef.com/tags/problems/dynamic-prog)[easy](http://www.codechef.com/tags/problems/easy)[witua](http://www.codechef.com/tags/problems/witua) |
| Date Added: | 20-03-2012 |
| Time Limit: | 4 sec |
| Source Limit: | 50000 Bytes |
| Languages: | ADA, ASM, BASH, BF, C, C99 strict, CAML, CLOJ, CLPS, CPP 4.3.2, CPP 4.8.1, CPP11, CS2, D, ERL, FORT, FS, GO, HASK, ICK, ICON, JAR, JAVA, JS, LISP clisp, LISP sbcl, LUA, NEM, NICE, NODEJS, PAS fpc, PAS gpc, PERL, PERL6, PHP, PIKE, PRLG, PYTH, PYTH 3.1.2, RUBY, SCALA, SCM guile, SCM qobi, ST, TCL, TEXT, WSPC |