CyberChallenge.IT 2026
Programming Test

Hamiltonian Grid [100 points]

Problem Statement

In a $R \times C$ grid ($R \leq 2$), numbered with $(0, 0)$ on the top left and $(R-1, C-1)$ on the bottom right, a non-negative integer is written in each cell, with the number on the $(0, 0)$ cell always being 0. A pawn is sitting in the $(0, 0)$ cell and, every second, it can do one of the following actions:

- Move horizontally or vertically, but not diagonally (i.e., incrementing or decrementing exactly one of its coordinates, when possible).
- Wait in that position without moving.

The pawn wants to visit each cell **exactly one time**, but it can visit a cell only if the number of seconds passed from the beginning is strictly higher than the number written in it. What is the minimum time required to visit all the cells?

Input

The input consists of the following lines:

- Line 1: an integer, T, representing the number of testcases.
- The following lines describe all T testcases. For each testcase:
  - The first line contains two space separated integers, R and C, representing the number of rows and columns of the grid.
  - The following R lines contain a space-separated list of C integers each. It is guaranteed that the first number of the first of these lines is 0.

Output

The output consists of T lines, each of them representing the answer to one of the testcases.

Scoring

Your program will be tested on a number of testcases grouped in subtasks. In order to obtain the score associated to a subtask, you need to correctly solve all its testcases.

- **Subtask 1** [**20 points**]: $1 \leq T \leq 100$, $R = 1$, $C \leq 10^3$
- **Subtask 2** [**40 points**]: $1 \leq T \leq 100$, $1 \leq R \leq 2$, $C \leq 10^3$
- **Subtask 3** [**40 points**]: $1 \leq T \leq 100$, $1 \leq R \leq 2$, $C \leq 10^5$

Examples

INPUT	OUTPUT
3 1 6 0 2 10 10 9 7 1 10 0 99 25 49 31 12 58 74 63 77 1 10 0 0 0 0 0 0 0 0 0 0	14 108 9

INPUT	OUTPUT
2 2 4 0 51 16 89 79 32 85 28 2 14 0 545 641 33 840 459 455 879 374 17 725 106 792 217 978 750 965 661 991 221 855 364 164 371 987 731 887 781	90 998

Explanation

- In the first testcase of the first input file, the pawn can move to the second cell at second 3, then it will have to wait until second 11 to move to the third one. Since 10 is the maximum of the grid, after it it will move every second, reaching the end at second 14.
- In the second testcase of the first input file, the pawn can move to the second cell at second 100. Since 99 is the maximum of the grid, after it it will move every second, reaching the end at second 108.
- In the third testcase of the first input file, the pawn can already move through all cells at second 1, reaching the end at second 9.
- In the first testcase of the second input file, the highest number is 89, which is on the top right corner of the grid, which will be therefore left as the last one to visit. The pawn can start moving down at second 80; then it will move right, up and right again without waiting. Then it will wait until second 86 to go down again, then right and it will finally wait until second 90 to visit the last cell and reach the end.
- In the second testcase of the second input file, the best strategy is to proceed straight to the right until the end of the grid, then go down and then head left until the last cell is visited. The pawn will wait until second 546 to visit the first cell, then until 642 for the next cell on the right; then it will go straight right one cell and wait again until 841 for the next one; then it will go straight right two cells in a row, waiting again until 880 for the following right cell. It will then be able to visit all cells in the first row without waiting, reaching the last cell on the top right corner at second 886. Then it will go down and then left two times, having to wait until second 988 to visit the following left cell. After that, it will just be able to proceed going left without waiting, reaching the end at second 998.