

Problem A

Dice Find Their Paths

Input File: *pa.in*
Time Limit: 1 second

Problem Description

We are rolling a standard die over a 2D grid with each cell being associated with a non-negative integer. As an example, the 4-by-4 array in Figure 1 is a 2D grid. A standard six-sided die is needed to traverse the grid (the layout of a standard six-sided die is shown in Figure 1 as well). You can arbitrarily place the die on one particular cell of the grid as the *starting position*. The rule requires the initial die configuration with the face “1” being placed on the bottom ; thus “6” is on the top face and “3” is facing the player (assume that the player is facing the grid from the bottom).

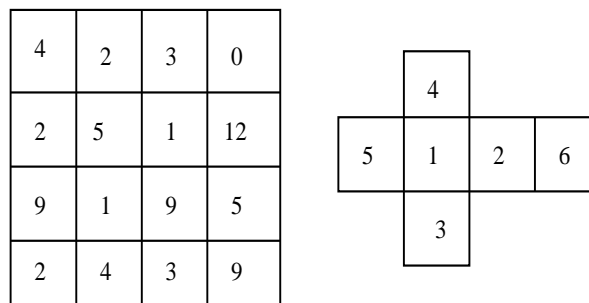


Figure 1: A grid configuration and the standard die.

To move through the grid you must tip the die over on an edge to land on an adjacent square, effecting horizontal or vertical movement from one square to another. However, there is a cost you must pay for such a move: one plus the square of the difference from the number of the landing face to the number marked on the cell. As an example, suppose that we move

the die on the starting position is row 3 and column 2 and tip the die over to its right edge. Then the resulting cost is just $1 + (2 - 9)^2 = 26$. On the other hand, if we tip the die over to its bottom edge, then the cost is $1 + (3 - 4)^2 = 2$.

There are many different paths for us to move the die off the starting cell to the destination. As an example, assume that the starting position is in (row 2, column 1) and the destination is (row 4, column 1). Let L (R) denote the move to the left (right), and let U (D) denote the move to the upper (down) cell. Then the cost of the straight line path (D, D) is $2 + (3 - 9)^2 + (4 - 2)^2 = 40$. Note that the longer path (R, D, D, L) has a lower cost $4 + (2 - 5)^2 + (3 - 1)^2 + (5 - 4)^2 + (1 - 2)^2 = 19$.

The goal of this problem is to write a program to find the cheapest path for the die to reach its destination. The input file will contain several grids for which the program should search for solutions. For each input grid, the program shall output the cost of the optimum path with lowest cost.

Input Format

The input consists of a list of k grids with each grid being associated with its own starting and destination position. The first integer (in a single line) represents the number of 2D grids, k . Note that k can be as large as 100. After the integer, there will be k grids with the following format: Each grid starts with a line containing six integers delimited by single spaces. These integers are, in order, the number of rows in the grid (an integer from 1 to 50, call this value R), the number of columns in the grid (an integer from 1 to 50, call this value C), the starting row, the starting column, the ending row, and the ending column. The next R lines contain C integers each, again delimited by single spaces. The integer associated with each cell of the grid can be as large as 20. This $R \times C$ array of integers defines the grid.

Output Format

For each grid (with its associated starting and ending position) output a single line containing the integer indicating the cost of the optimum path with lowest cost.

Sample Input

```
3
4 4 2 1 4 1
4 2 3 0
2 5 1 12
9 1 9 5
2 4 3 9
2 2 1 1 2 2
0 3
2 5
1 1 1 1 1 1
4
```

Sample Output

```
19
7
0
```