

Rock Climbing

This question is graded for 1.5%!

Statement

Fluffy the Hamster is trying out (indoor) rock climbing to get fit, on the advice of his malamute friend. Fluffy being lazy, he wants to do things with the **minimum** amount of effort.

The wall can be represented as a $W \times H$ 2D grid (W columns, H rows), each cell representing a temporary resting spot. Fluffy starts at the bottom-left cell $(0, 0)$, and his goal is the top-right cell $(W - 1, H - 1)$.

The face of the rock climbing wall is pretty uneven, so moving around is easier in some places, compared to other places on the wall.

- When moving between horizontally adjacent pair of cells $(x, y) \leftrightarrow (x + 1, y)$ in either direction, there exists a sideways handhold that Fluffy uses. It takes $S_{x,y}$ amount of effort, to grip this handhold and move across.
- Similarly, when climbing up from $(x, y) \rightarrow (x, y + 1)$, Fluffy expends $U_{x,y}$ amount of effort to pull himself up, using a vertical handhold.
- However, when going downward, Fluffy *always* takes a risk by doing a blind leap down. When jumping down d cells, he spends 1 effort **per vertical step travelled** safely falling down, and 1 effort to catch himself at the target cell. (Note that this does not depend on which cell Fluffy is currently in.)

Fluffy cannot exit the grid (e.g. he cannot move left when in the leftmost column, nor drop down when in the bottom-most row). Movement within a cell takes negligible effort.

In summary, from his current cell (x, y) , he has (at most) 4 ways to move:

1. Fluffy can move one cell left $(x, y) \rightarrow (x - 1, y)$, at the cost of $S_{x-1,y}$ effort.
2. Fluffy can move one cell right $(x, y) \rightarrow (x + 1, y)$, at the cost of $S_{x,y}$ effort.
3. Fluffy can climb up one cell $(x, y) \rightarrow (x, y + 1)$, at the cost of $U_{x,y}$ effort.
4. Fluffy can leap down **multiple** cells $(x, y) \rightarrow (x, y - d)$, at the cost of $d + 1$ effort.

The total amount of effort is the sum of effort costs over his entire journey. Can you help Fluffy find the least amount of effort he needs to spend climbing the wall?

Constraints

- $1 \leq W, H \leq 500$.
- $0 \leq S_{x,y}, U_{x,y} < 1000$.
- All effort costs are non-negative integers.

Input

The first line of input contains 2 integers, H W , the height and the width of the grid.

The next H lines will describe a $H \times (W - 1)$ matrix of $S_{x,y}$, the sideways handhold effort costs. The first such line describes the bottom-most row ($y = 0$), all the way til the last such row describing the topmost row $y = H - 1$. Each line contains $W - 1$ integers $S_{x,y}$, the costs for sideways handholds in the x -th row going left-to-right.

Afterwards, yet another $H - 1$ lines describe the $(H - 1) \times W$ matrix of $U_{x,y}$, the upward handhold effort costs. The first such line describes the bottom-most row ($y = 0$), all the way til the last such row describing the second-topmost row $y = H - 2$. Each line contains W integers $U_{x,y}$, the costs for the upward handholds of the x -th row going left-to-right.

Output

Print a single integer, the minimum amount of effort Fluffy needs to get from the bottom-left corner to the top-right corner of the wall.

Examples

Sample Input	Expected Output
<pre> 5 5 3 1 0 1 9 1 1 2 9 8 2 3 1 3 2 4 4 9 4 2 6 6 9 5 6 8 0 7 2 9 1 7 0 4 2 0 0 7 10 4 </pre>	<pre> 20 </pre>

Notes

1. A skeleton file has been given to help you. You should not create a new file or rename the file provided. You should develop your program using this skeleton file.
2. You are free to define your own helper methods and classes (or remove existing ones) if it is suitable but you must put all the new classes, if any, in the same skeleton file provided.

Skeleton File

You are given the skeleton file `RockClimbing.java`. You should see the following contents when you open the file:

```
/**
 * Name      :
 * Matric. No :
 */

import java.util.*;

public class RockClimbing {
    private void run() {
        // implement your "main" method here
    }

    public static void main(String args[]) {
        RockClimbing runner = new RockClimbing();
        runner.run();
    }
}
```