

Problem J. Joy of Tracking

Input file: *standard input*
Output file: *standard output*
Time limit: 3 seconds
Memory limit: 1024 mebibytes

You are a big fan of a particular feature of an online map service: route tracking. You enjoy drawing pictures on the online map: first, design a route looking like the desired image, and then actually trace the route with your mobile device.

One day, you noticed the town has a grid-shaped train network. On the map, there are $H \times W$ stations on a grid with H horizontal lines and W vertical lines. Each crossing point has exactly one station, and each station is connected to all the stations adjacent to it vertically or horizontally (but not diagonally). Connections can have different fees, but the fee to move from station A to station B is always the same as the fee to move from B to A . If you use a connection multiple times, you have to pay the fee each time you use it.

You plan to draw a complete grid on the map by going through all the connections on the train network at least once. You have to start and finish the route at the same station. Under these constraints, you want to minimize the total cost of travel. As you are also good at programming, you decided to write a program to calculate the minimum cost when you design an optimal route.

Input

The first line contains two integers H ($2 \leq H \leq 100$) and W ($2 \leq W \leq 100$) which represent that the train network grid consists of H rows and W columns. Let (i, j) be the crossing at the i -th row from the top and the j -th column from the left.

The i -th of the following $H - 1$ lines contains W integers, where the j -th integer is the fee to move between the station at (i, j) and the station at $(i + 1, j)$.

The i -th of the following H lines contains $W - 1$ integers, where the j -th integer is the fee to move between the station at (i, j) and the station at $(i, j + 1)$.

All the fees are at least 0 and at most 10^9 .

Output

Output a line with a single integer: the minimum cost to draw a complete grid by taking trains on the train network.

Examples

<i>standard input</i>	<i>standard output</i>
2 4 2 2 2 2 1 1 1 1 1 1	16
4 3 3 2 0 6 1 0 7 1 6 7 5 8 1 8 3 3 5	76