Best spot

In Chile, land are partitioned into a one large grid, where each element represents a land of size *1x1*.

Shaka is a newcomer in Chile and is trying to start his own business. He is planning to build a store. He has his own ideas for the "perfect store" which can be represented by a *HxW* grid. Element at position *(i, j)* represents height of land at index *(i, j)* in the grid.

Shaka has purchased a land area which can be represented *RxC* grid (*H <= R, W <= C*). Shaka is interested in finding best *HxW* sub-grid in the acquired land. In order to compare the possible sub-grids, Shaka will be using the sum of squared difference between each cell of his "perfect store" and it's corresponding cell in the subgrid. Amongst all possible sub-grids, he will choose the one with smallest such sum.

Note

- The grids are 1-indexed and rows increase from top to bottom and columns increase from left to right.
- If *x* is the height of a cell in the "perfect store" and *y* is the height of the corresponding cell in a sub-grid of the acquired land, then the squared difference is defined as $(x-y)^2$

Input Format
The first line of the input consists of two integers, *R C*, separated by single space.
Then *R* lines follow, each one containing *C* space separated integers, which describe the height of each land spot of the purchased land.
The next line contains two integers, *H W*, separated by a single space, followed by *H* lines with *W* space separated integers, which describes the "perfect store".

Output Format

In the first line, output the smallest possible sum (as defined above) Shaka can find on exploring all the sub-grids (of size *HxW*) in the purchased land.
In second line, output two space separated integers, *i j*, which represents the index of top left corner of sub-grid (on the acquired land) with the minimal such sum. If there are multiple sub-grids with minimal sum, output the one with the smaller row index. If there are still multiple sub-grids with minimal sum, output the one with smaller column index.

Constraints

1 <= *R, C* <= 500
1 <= *H* <= *R*
1 <= *W* <= *C*
No height will have an absolute value greater than 20.

Sample Input:

```
3 3
19 19 -12
5 8 -14
-12 -11 9
2 2
-18 -12
-10 -7
```

Sample Output:

```
937
2 2
```

Explanation

The result is computed as follows: $(8 - (-18))^2 + (-14 - (-12))^2 + (-11 - (-10))^2 + (9 - (-7))^2 = 937$