

Problem B. Luxury burrow

Input file: `burrow.in`
Output file: `burrow.out`
Time limit: 2 seconds
Memory limit: 64 megabytes

Hobbit Bilbo is very tired from his adventures and decided to build a new burrow. The hill, where he plans to buy some land, has a shape of a rectangle and it can be represented as a table consisting of N rows and M columns (rows and columns are numbered starting from 1, where 1 corresponds to the topmost row and the leftmost column). Each cell of the table corresponds to a 1×1 square piece. Hobbits like simple shapes, that's why Bilbo intends to buy a plot of land of a rectangular shape, with sides parallel to the sides of the hill. In other words he chooses x_1, x_2, y_1, y_2 ($x_1 \leq x_2, y_1 \leq y_2$) and buys all cells (x, y) , such that $(x_1 \leq x \leq x_2)$ and $(y_1 \leq y \leq y_2)$.

After all adventures he took part in, Bilbo is very rich and is not worried about the total price. However, he thinks a lot about his reputation and thus the price of the cheapest cell he will buy must be as high as possible. He also wants his new burrow to be enough to fit all of his furniture, that means the square of a rectangular plot he will buy has to have an area greater than or equal to K . If there are multiple possible plots, Bilbo chooses the one with the largest area.

Bilbo asks you to find the best plot of land for him.

Input

The first line of input file contains three positive integers N, M and K , divided by exactly one space — the number of rows, the number of columns and the minimal possible square of land plot Bilbo can live on, respectively. Each of the following N lines contains M numbers — the prices of the corresponding cells. Hence, the number j in the line $i + 1$, corresponds to the cell (i, j) . K does not exceed the total number of cells in the table, all prices belong to range from 1 to 10^9 , inclusive.

Output

As the answer, output two numbers separated by space. First goes the price of the cheapest cell in the resulting rectangular plot. Second is the total square of this plot.

Examples

burrow.in	burrow.out
3 3 3 1 1 1 1 2 2 1 2 2	2 4
1 10 5 4 3 2 5 10 7 6 5 1 100	5 5
3 5 2 5 7 5 5 5 8 5 5 7 5 8 5 8 8 8	8 3

Note

This task has 3 subtasks. Only full solution of a subtask will be counted.

Subtask 1. (23 points) $1 \leq N, M \leq 200$.

Subtask 2. (33 points) $1 \leq N, M \leq 400$.

Subtask 3. (44 points) $1 \leq N, M \leq 1000$.