Farmers Cattle

Farmer john owns a single cow and he loves it a lot. The cow has a disease and is going to die. To survive, the cow needs medicine of a particular type each day. Let us say the cow needs **medicine**[i] to survive the i^{th} day. (medicine[i] will be terminated by -1, which is an unavailable medicine, and the cow has to invariably die that day).

To help the cow, john has decided to buy pastures of some medical value. Farmer sees a two-dimensional grid of pastures, each cell having exactly one medical herb. Now he needs to buy a sub-rectangular region of the grid, whose area cannot exceed **A** (**A** > 1). With this region the farmer intends to feed his cow, as long as possible.

Input Format:
The input file consists of multiple testcases.
The first line of each testcase contains three integers, **R**, **C** and **A**.
The second line consists of sequence of integers describing **medicine**[i]. This list will be terminated by -1.
The next **R** lines contain **C** integers each, specifying the medicinal type of the herb in that cell. (1 <= **R,C** <= 200). All herbs are specified by non negative integers.
Input terminates with a line containing three zeros and must not be processed.

Output Format:
For each testcase print a single line containing 5 integers:
days r1 c1 r2 c2
(1 <= r1 <= r2 <= R, 1 <= c1 <= c2 <= C)

- ***days*** is the number of days the cow survives. We wish to maximise this.
- If there are more than one solutions print the one with minimal r1.
- If there are more than one solutions still, print the one with minimal c1.
- If there are more than one solutions still, print the one with minimal r2.
- If there are more than one solutions still, print the one with minimal c2.

Sample Input:

```
3 4 6
12 30 12 100 22 -1
30 12 5 3
12 30 100 5
22 3 22 100
3 4 6
2 30 12 100 22 -1
30 12 5 3
12 30 100 5
22 3 22 100
3 4 6
12 30 12 100 22 -1
30 12 5 3
12 30 100 5
22 12 22 100
```

0 0 0

Sample Output:

4 1 1 2 3
0 1 1 1 1
5 1 2 3 3