D. Constellation

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

A star map in Berland is a checked field $n \times m$ squares. In each square there is or there is not a star. The favourite constellation of all Berland's astronomers is the constellation of the Cross. This constellation can be formed by any 5 stars so, that for some integer x (radius of the constellation) the following is true:

- the 2nd is on the same vertical line as the 1st, but x squares up
- the 3rd is on the same vertical line as the 1st, but x squares down
- the 4th is on the same horizontal line as the 1st, but x squares left
- the 5th is on the same horizontal line as the 1st, but x squares right

Such constellations can be very numerous, that's why they are numbered with integers from 1 on the following principle: when two constellations are compared, the one with a smaller radius gets a smaller index; if their radii are equal — the one, whose central star if higher than the central star of the other one; if their central stars are at the same level — the one, whose central star is to the left of the central star of the other one.

Your task is to find the constellation with index k by the given Berland's star map.

Input

The first line contains three integers n, m and k ($1 \le n, m \le 300$, $1 \le k \le 3 \cdot 10^7$) — height and width of the map and index of the required constellation respectively. The upper-left corner has coordinates $(1, 1)$, and the lower-right — (n, m). Then there follow n lines, m characters each — description of the map. j-th character in i-th line is «*», if there is a star in the corresponding square, and «.» if this square is empty.

Output

If the number of the constellations is less than k, output `-1`. Otherwise output 5 lines, two integers each — coordinates of the required constellation. Output the stars in the following order: central, upper, lower, left, right.

Examples

input

```
5 6 1
....*.
...***
....*.
..*...
.***..
```

output

```
2 5
1 5
3 5
2 4
2 6
```

input

```
5 6 2
....*.
...***
....*.
..*...
.***..
```

output

```
-1
```

input

```
7 7 2
...*...
.......
...*...
*.***.*
...*...
.......
...*...
```

output

```
4 4
1 4
7 4
4 1
4 7
```