H. New Year and Snowy Grid

time limit per test: 9 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Pay attention to the output section below, where you will see the information about flushing the output.

Bearland is a grid with h rows and w columns. Rows are numbered 1 through h from top to bottom. Columns are numbered 1 through w from left to right. Every cell is either allowed (denoted by '`.`' in the input) or permanently blocked (denoted by '`#`').

Bearland is a cold land, where heavy snow often makes travelling harder. Every day a few allowed cells are **temporarily** blocked by snow. Note, that this block works only on this particular day and next day any of these cells might be allowed again (unless there is another temporarily block).

It's possible to move directly between two cells only if they share a side and none of them is permanently or temporarily blocked.

Limak is a little polar bear who lives in Bearland. His house is at the top left cell, while his school is at the bottom right cell. Every day Limak should first go from his house to the school and then return back to his house. Since he gets bored easily, he doesn't want to **visit the same cell twice** on one day, except for the cell with his house, where he starts and ends. If Limak can reach a school and return home avoiding revisiting cells, he calls a day *interesting*.

There are q days you must process, one after another. For each of these days you should check if it's interesting and print "`YES`" or "`NO`" on a separate line. In order to be able to read the description of the next day you should print the answer for the previous one and flush the output.

It's guaranteed that a day with no cells temporarily blocked by snow would be interesting. It's also guaranteed that cells with Limak's house and school are never blocked (neither permanently or temporarily).

Input

The first line of the input contains three integers h, w and q ($2 \le h, w \le 1000$, $1 \le q \le 10\,000$) — the height and the width of the grid, and the number of days, respectively.

Next h lines describe which cells are allowed and which permanently blocked. The i-th line contains a string of length w, describing the i-th row. Every character is either '`.`' (denoting an allowed cell) or '`#`' (denoting a permanently blocked cell). It's guaranteed that a day with no cells temporarily blocked by snow would be interesting.

Then, the description of q days is given. The description of the i-th day starts with a line containing a single integer k_i ($1 \le k_i \le 10$) — the number of cells that are temporarily blocked by snow on that day. Each of next k_i lines contains two integers $r_{i,j}$ and $c_{i,j}$ ($1 \le r_{i,j} \le h$, $1 \le c_{i,j} \le w$), representing a cell at the intersection of the row $r_{i,j}$ and the column $c_{i,j}$. The given k_i cells are distinct and none of them is permanently blocked. Also, none of them contains Limak's house or school.

Output

For each of q days print "`YES`" if that day is interesting, and otherwise print "`NO`", both without the quotes. After printing an answer, you have to both print the end-of-line character and `flush` the output. Then you can proceed to the next day. You can get `Idleness Limit Exceeded` if you don't print anything or if you forget to flush the output.

To flush you can use (just after printing a `YES/NO` and end-of-line):

- `fflush(stdout)` in C++;
- `System.out.flush()` in Java;
- `stdout.flush()` in Python;
- `flush(output)` in Pascal;

- See the documentation for other languages.

Examples

input

```
3 5 4
.....
.....
.#...
1
1 4
1
1 5
2
2 4
3 1
2
1 5
3 3
```

output

```
NO
YES
YES
NO
```

input

```
9 31 5
...............................
...............................
.###.###.#.###...###.###.#.###.
...#.#.#.#.#.......#.#.#.#...#.
.###.#.#.#.###...###.#.#.#...#.
.#...#.#.#.#.#...#...#.#.#...#.
.###.###.#.###...###.###.#...#.
...............................
...............................
5
6 5
2 11
1 14
8 15
2 14
5
2 14
1 14
8 16
6 5
2 11
3
2 2
1 4
8 30
10
3 1
3 11
5 16
7 21
4 16
3 5
7 31
3 9
7 25
3 27
10
3 1
3 9
```

```
7 25
3 27
7 21
4 17
3 5
7 31
4 16
3 11
```

output

```
NO
YES
YES
YES
NO
```

Note

In the first sample, there are 4 days. Drawings below show how Limak could go to school and return to his home in the second and the third day (on the left and on the right respectively). A permanently blocked cell is painted red, while cells temporarily blocked by snow are painted orange. Black and green arrows should Limak's way to the school and back to the house respectively.

For the second sample, below you can see how the grid looks like on each day, where '`#`' denotes a cell that is blocked, either temporarily or permanently.