

## E. Wilbur and Strings

time limit per test: 3 seconds  
memory limit per test: 256 megabytes  
input: standard input  
output: standard output

Wilbur the pig now wants to play with strings. He has found an  $n$  by  $m$  table consisting only of the digits from 0 to 9 where the rows are numbered 1 to  $n$  and the columns are numbered 1 to  $m$ . Wilbur starts at some square and makes certain moves. If he is at square  $(x, y)$  and the digit  $d$  ( $0 \leq d \leq 9$ ) is written at position  $(x, y)$ , **then he must** move to the square  $(x + a_d, y + b_d)$ , if that square lies within the table, and he stays in the square  $(x, y)$  otherwise. Before Wilbur makes a move, he can choose whether or not to write the digit written in this square on the white board. All digits written on the whiteboard form some string. Every time a new digit is written, it goes to the end of the current string.

Wilbur has  $q$  strings that he is worried about. For each string  $s_i$ , Wilbur wants to know whether there exists a starting position  $(x, y)$  so that by making finitely many moves, Wilbur can end up with the string  $s_i$  written on the white board.

### Input

The first line of the input consists of three integers  $n, m$ , and  $q$  ( $1 \leq n, m, q \leq 200$ ) — the dimensions of the table and the number of strings to process, respectively.

Each of the next  $n$  lines contains  $m$  digits from 0 and 9 giving the table itself.

Then follow 10 lines. The  $i$ -th of them contains the values  $a_{i-1}$  and  $b_{i-1}$  ( $-200 \leq a_i, b_i \leq 200$ ), i.e. the vector that Wilbur uses to make a move from the square with a digit  $i-1$  in it.

There are  $q$  lines that follow. The  $i$ -th of them will contain a string  $s_i$  consisting only of digits from 0 to 9. It is guaranteed that the total length of these  $q$  strings won't exceed 1 000 000.

### Output

For each of the  $q$  strings, print "YES" if Wilbur can choose  $x$  and  $y$  in order to finish with this string after some finite number of moves. If it's impossible, than print "NO" for the corresponding string.

### Examples

#### input

```
1 1 2
0
1 1
1 1
1 1
1 1
1 1
1 1
1 1
1 1
1 1
1 1
1 1
00000000000000
2413423432432
```

#### output

```
YES
NO
```

#### input

```
4 2 5
01
23
45
```

```
67
0 1
0 -1
0 1
0 -1
0 1
0 -1
0 1
0 -1
0 1
0 -1
0 1
0000000000
010101011101
32232232322
44343222342444324
6767
```

output

```
YES
YES
YES
NO
YES
```

Note

In the first sample, there is a 1 by 1 table consisting of the only digit 0. The only move that can be made is staying on the square. The first string can be written on the white board by writing 0 repeatedly. The second string cannot be written as there is no 2 on the table.