

F. Race

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

The Old City is a rectangular city represented as an $m \times n$ grid of blocks. This city contains many buildings, straight two-way streets and junctions. Each junction and each building is exactly one block. All the streets have width of one block and are either vertical or horizontal. There is a junction on both sides of each street. We call two blocks adjacent if and only if they share a common side. No two blocks of different streets are adjacent and no two junctions are adjacent.

There is an annual festival and as a part of it, The Old Peykan follows a special path in the city. This path starts from a block in a street, continues with many junctions and ends in a block of some street. For each street block, we know how much time it takes for the Old Peykan to go from this block to an adjacent block. Also the Old Peykan can go from each junction to its adjacent street blocks in one minute. Of course Old Peykan can't go to building blocks.

We know the initial position of the Old Peykan and the sequence of junctions that it passes to reach its destination. After passing all the junctions and reaching the destination, it will stay there forever. Your task is to find out where will the Old Peykan be k minutes after it starts moving. Consider that The Old Peykan always follows the shortest path that passes through the given sequence of junctions and reaches the destination.

Note that the Old Peykan may visit some blocks more than once.

Input

The first line of input contains three integers m , n and k ($3 \leq m, n \leq 100$, $1 \leq k \leq 100000$). Next m lines are representing the city's map. Each of them contains n characters, each character is a block:

- Character "#" represents a building.
- Digits "1", "2", ..., "9" represent a block of a street and this digit means the number of minutes it takes for the Old Peykan to pass this block.
- Characters "a", "b", ..., "z" means that this block is a junction and this character is its name. All the junction names are unique.

Consider that all blocks have the coordinates: the j -th in the i -th line have coordinates (i, j) ($1 \leq i \leq m$, $1 \leq j \leq n$).

The $(m + 2)$ th line contains two integers r_s and c_s ($1 \leq r_s \leq m$, $1 \leq c_s \leq n$), string s and another two integers r_e and c_e ($1 \leq r_e \leq m$, $1 \leq c_e \leq n$). The path starts from block (r_s, c_s) , continues through junctions in the order that is specified by s and will end in block (r_e, c_e) . Length of s is between 1 and 1000.

It's guaranteed that string s denotes a correct path from the start position to the end position and string s doesn't contain two consecutive equal letters. Also start position (r_s, c_s) and the end position (r_e, c_e) are street blocks.

Output

In a single line print two integers r_f and c_f — (r_f, c_f) being the position of the Old Peykan after exactly k minutes.

Examples

input
3 10 12 ##### #z1a1111b# ##### 2 3 ab 2 8
output
2 8

input

10 3 5

#w#
#1#
#a#
#1#
#1#
#1#
#1#
#b#

3 2 abababababababab 6 2

output

8 2

input

3 10 6

#z1a1311b#

2 3 ab 2 8

output

2 7