Zigzagging Dwarf Trains (z)

Memory limit: 1024 MB Time limit: 1.00 s

Dwarf the Dispatcher is in charge of organizing rat-pulled trains supplying the underground dwarf city. Each train consists of different types of carriages carrying different types of goods. Carriage types are denoted by single lowercase letters from the English alphabet. There are N trains to be organized according to a given plan, and all trains have equal length L (rats are smart and refuse to pull trains longer than others). The trains are initially stationed on N sidetracks, one train per track.

The Dispatcher can detach a carriage from the end of any train and attach it to the end of any other train. However, all sidetracks have limited length M, so each train can have at most M carriages during the rearrangement. Help the dwarf rearrange the trains: given the initial and target configurations, determine whether it is possible to rearrange the trains without exceeding the maximum length.

Input

The first line of input contains three positive integers N, M, and L: the number of trains, the maximum length of a train during the rearrangement, and the (initial and final) length of all trains, respectively.
The next N lines describe the initial train configurations, one per line. Each line contains a string of L lowercase English letters describing the carriage types in the train from beginning to end.
The next N lines describe the target train configurations in the same format.

Output

In the first line, print `YES` if it is possible to achieve the target configuration, or `NO` otherwise.
If the answer is `YES`, in the next line print S being the number of steps needed to achieve the target configuration. Each of the following S lines should contain two positive integers a and b, indicating that a carriage is moved from the end of train a to the end of train b. It is not necessary to minimize S, but it must not exceed 10^6.

Limits

$1 \leq N \leq 100$,
$1 \leq L \leq M \leq 100$,
$N \cdot L \leq 100$.

Examples

Input	Output
3 3 1 a b c c a b	YES 5 1 2 3 1 2 1 2 3 1 2

Input

```
3 5 3
abc
dee
fff
cbe
dff
fae
```

Output

```
YES
17
1 2
1 3
1 3
2 1
3 1
1 2
3 1
2 1
1 3
2 1
3 1
2 1
3 2
3 2
1 2
1 3
2 3
```

Input

```
2 3 2
aa
bc
aa
cb
```

Output

```
NO
```