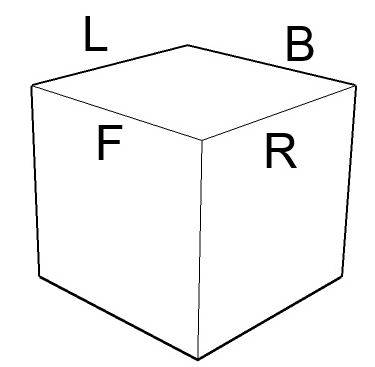
## Problem 3 – Slides

You are given a **rectangular cuboid** of size **W** (width), **H** (height) and **D** (depth) consisting of **W** \* **H** \* **D** cubes. The top row (layer) of the cube is considered to be with height 0. Each cube can contain one of the following things – slide, teleport, basket or just be an empty cube. A small ball that can fit through a single cube is dropped over the cuboid. The ball can pass through cubes according to some rules and can exit the cuboid only through the bottom side, but not through the walls. Your task is to find if the ball can exit the cuboid and also the coordinates of the final cube that it got to (either before exiting or when stuck somewhere). The elements of the cuboid are given as strings:

* Slide – The ball slides on it to the next level of the cuboid in the given direction. Each slide is given in the following pattern “S X” where X is the direction. The directions are: “L” – left, “R” – right, “F” – front, “B” – back, “FL” – front left, “FR” – front right, “BL” – back left, “BR” – back right. E.g. if the ball is in cube [2, 0, 0] with slide (S BL) it slides to cube[1, 1, 1]. If the ball can’t slide in the given direction (i.e. to a wall), it’s stuck and can’t exit the cuboid. If the ball is on a slide on the last row it is considered that it can slide and exit regardless of the direction.
* Teleport – Surprisingly the ball gets teleported to another cube on the same row. Each teleport is given in the following pattern “T W D” where “W” and “D” are the width and the depth of the destination cube e.g. (T 1 2) Two teleports will not point to each other.
* Empty cube – The ball falls through it and goes directly to the next level. It is given as a single letter “E” e.g. (E)
* Basket – If the ball falls in a basket it’s stuck there and can’t exit. The basket is given as single letter “B” e.g. (B)

### Input

The input data should be read from the console. At the first line 3 integers **W**, **H** and **D** are given separated by a space. These numbers specify the width, height and depth of the cuboid. At the next **H** lines the cubes are given as **D** sequences of exactly **W** strings each one in brackets “( )”. Each sequence of **W** strings is separated from the next one by " **|** " (space + vertical line + space). At the last line there will be two integer numbers **ballW** and **ballD** that specify the location where the ball is initially dropped.

The input data will be correct and there is no need to check it explicitly.

### Output

The output data should be printed on the console.

On the first output line you should print “Yes” if the ball can exit the cuboid or “No” otherwise.

On the second line print the 3 coordinates (w, h, d) of the last cube that the ball got to.

### Constraints

* The numbers **W**, **H** and **D** are all integers in the range [3…50].
* The numbers **ballW** and **ballD** specify the indexes in the cuboid array.
* 0 ≤ ballW < W, 0 ≤ ballD < D
* Allowed working time for your program: 0.3 seconds. Allowed memory: 16 MB.

### Examples

|  |  |
| --- | --- |
| **Example input** | **Example output** |
| 3 3 3  (S L)(E)(S L) | (S L)**(S R)**(S L) | (B)(S F)(S L)  (S B)(S F)(E) | (S B)**(S F)(T 1 1)** | (S L)(S R)(B)  (S FL)**(S FL)**(S FR) | (S FL)(S FL)(S FR) | (S F)(S BR)(S FR)  1 1 | Yes  1 2 0 |

|  |  |
| --- | --- |
| **Example input** | **Example output** |
| 3 3 3  (S L)**(E)**(S L) | (S L)(S R)(S L) | (B)(S F)(S L)  (S B)**(S R)**(E) | (S B)(S F)(T 1 1) | (S L)(S R)(B)  (S FL)(S FL)**(B)** | (S FL)(S FL)(S FR) | (S F)(S BR)(S FR)  1 0 | No  2 2 0 |