

You are given a map of a labyrinth, and your task is to find a path from start to end. You can walk left, right, up and down. If you are on the left column of the labyrinth and you go left, you will wrap around and be on the same row but on the last column to the right, unless there is a wall on that location in which case you cannot move. Similarly if you are on the last column to the right and move right you will wrap around and become located on the same row but the first column to the left, unless there is a wall on that location. Same wrap around rules are applied also for the vertical motion up and down.

## Input

The first input line has two integers  $n$  and  $m$ : the height and width of the map.

- $1 \leq n, m \leq 1000$

Then there are  $n$  lines of  $m$  characters describing the labyrinth. Each character is . (floor), # (wall), A (start), or B (end). There is exactly one A and one B in the input.

## Output

First print "YES", if there is a path, and "NO" otherwise.

If there is a path, print the length of the shortest such path.

## Constraints

### Example

Input:

```
5 8
#####
#.A#...#
#.#.#B#
#.....#
#####
```

Output:

```
YES
9
```

Input:

```
5 8
#####
..A#....
#.#.#B#
#.....#
#####
```

Output:

```
YES
5
```

Input:

```
5 8
.#####
A..#...#
#.#.#.#
#.....B#
.....
```

Output:

```
YES
5
```