# Problem K. Knight

| | |
|---|---|
| Input file: | *standard input* |
| Output file: | *standard output* |
| Time limit: | 2 seconds |
| Memory limit: | 512 mebibytes |

There is a chessboard with $n$ rows and $m$ columns. Some squares on the chessboard are broken. There are two knights on the chessboard, controlled by Alice and Bob. The movement of a knight is determined by two parameters $r$ and $c$. On each step, Alice or Bob can move their knight to a square which is $r$ squares away horizontally and $c$ squares vertically, or $r$ squares away vertically and $c$ squares horizontally.

Alice and Bob take turns playing, starting with Alice. On each turn, the player moves his or her knight. However, the player can not move the knight to a square which is broken or is occupied by the other knight.

There is an extra constraint. The configuration of the knights can be viewed as an ordered pair $(a, b)$ where $a$ is Alice's square and $b$ is Bob's square. It is forbidden to repeat a configuration which already occurred earlier.

A player loses if he or she can not make a move on his or her turn. Determine the winner if both players play optimally.

## Input

The first line contains four integers $n$, $m$, $r$, and $c$ ($1 \le n, m \le 1000$, $0 \le r < n$, $0 \le c < m$).

Each of the following $n$ lines contains a string of length $m$. Together, these lines describe the chessboard. There are four types for each square:

- "`@`": The square is broken.
- "`.`": The square is not broken.
- "`A`": The square is not broken. It is the start position of Alice's knight.
- "`B`": The square is not broken. It is the start position of Bob's knight.

It is guaranteed that the squares "`A`" and "`B`" both occur exactly once on the chessboard.

## Output

Output the name of the winner: "`Alice`" or "`Bob`".

## Example

| standard input | standard output |
|---|---|
| `2 3 1 2`<br>`A@.`<br>`B@.` | `Alice` |

## Note

On the first step, Alice moves the knight to the square $(2, 3)$.

On the second step, Bob moves the knight to the square $(1, 3)$.

On the third step, Alice moves the knight back to the square $(1, 1)$.

On the fourth step, Bob can not move the knight back to the square $(2, 1)$, because it will create the ordered pair of squares $(1, 1), (2, 1)$ which is the same as the position in the beginning. Alice wins.