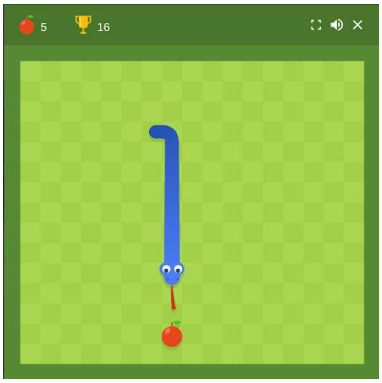
Problem L
Snake
Time limit: 2 seconds

Snake is a video game classic, preserved at the Museum of Modern Art (MoMA) and listed as one of the "Top 100 Video Games" of all time. The goal of the game is to move a snake's head to an apple. Once the snake reaches the apple, it eats it and grows in length. A new apple is placed, which the now grown snake must then eat.



Google's version of Snake

The game is played on a grid, and every segment of the snake's body occupies one cell. The snake's head can turn in three directions, but it cannot go backwards. The body follows the head. The head may not collide with the body or exit the grid. Since the entire snake moves at the same time, the head is allowed to enter the cell that the tail is vacating.

Playing the game requires quickness and foresight. It's all too easy to take turns that put the snake head in a position where it's doomed to hit the wall or its body before reaching the apple, especially as the snake grows longer.

You're being asked to write a program that can determine whether the snake's head can reach the apple from a given position, or not and the snake is doomed to die.

Input

The first line of output contains two integers r and c ($1 \le r, c \le 10, r \cdot c \ge 2$), where the grid has r rows and c columns.

Each of the next r lines contains a string of length exactly c characters from the set

{'.','0', ... ,'9','a', ... ,'f','A'}

where '.' represents an open cell in the grid, the hexadecimal digits '0', ... ,'9' and 'a', ... ,'f' represent the snake, and 'A' represents the apple. The snake may be anywhere from one to sixteen characters long, with '0' as its head, followed by the other hexadecimal digits in strict order ('1' follows '0', '2' follows '1', etc., with no skipping digits.). It is guaranteed that there is at most one of each digit, each digit (except '0') is adjacent to the immediately previous digit, and that there is exactly one apple in the grid.

Output

Output a single integer, which is 1 if the snake can reach the apple, and 0 if it cannot and is doomed to die.

Sample Input 1

```
5 8
......01
....98.2
...A.7.3
.....654
........
```

Sample Output 1

```
1
```

Sample Input 2

```
5 4
...A
....
6789
5432
..01
```

Sample Output 2

```
0
```

Sample Input 3

```
5 5
....A
.....
678..
54321
....0
```

Sample Output 3

```
1
```

Sample Input 4

```
4 4
567A
4389
12ba
0dc.
```

Sample Output 4

```
1
```