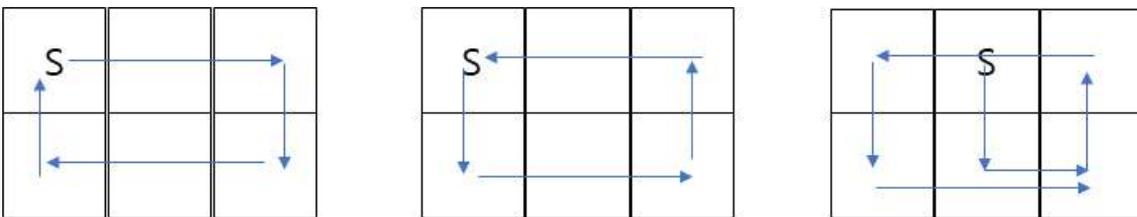


문제4 : 칠하기

$N \times M$ 칸으로 이루어진 격자가 있다. 격자의 일부 칸들은 막혀서 갈 수 없는 칸들이고, 다른 칸들은 갈 수 있는 칸들이다. 이제 다음 규칙을 따라서 갈 수 있는 칸들에 색을 칠하려고 한다.

1. $N \times M$ 칸 중 한 칸을 현재 칸으로 잡고 시작한다. 시작하는 현재 칸은 갈 수 있는 칸 중 어떤 것이든 가능하다.
2. 현재 칸에서 한 방향(위/아래/왼쪽/오른쪽 중 하나)을 정하고, 격자의 왼쪽/오른쪽/위/아래 경계에 도착하거나 이동하려는 칸이 막혀 있을 때까지 **계속** 이동한다. 중간에 멈출 수 없음에 유의하시오.
3. 정지할 때까지 만약 가로 방향(왼쪽 또는 오른쪽)으로 이동했다면 가로로 이동했던 칸들이 모두 노란 색으로 칠해진다. 이동을 시작한 칸도 노란 색으로 칠해진다. 만약 정지할 때까지 세로 방향(위 또는 아래)으로 이동했다면 세로로 이동했던 칸들이 모두 파란 색으로 칠해진다. 이동을 시작한 칸도 파란 색으로 칠해진다. (이동하려는 방향이 바로 막혀 있어서 한 칸도 이동할 수 없었던 경우에도 시작하는 칸에 색이 칠해지는 것으로 정한다.)
4. 모든 갈 수 있는 칸을 적어도 한 번 이상 노란 색으로 칠했고, 또 적어도 한 번 이상 파란 색으로 칠했다면 성공이며 종료한다.
5. 만약 그렇지 않다면, 2번 과정이 끝났을 때 정지한 칸, 즉 마지막으로 이동할 수 있었던 칸으로부터 시작하여 다시 2-4번 과정을 반복한다. 이 과정을 무한히 반복하더라도 모든 갈 수 있는 칸을 적어도 한 번 이상 노란색으로, 또 적어도 한 번 이상 파란 색으로 칠할 방법이 없다면 실패이다.

예를 들어 다음 예를 살펴보자. 2×3 격자의 모든 칸이 갈 수 있는 칸들인 다음 예제에서, 가장 왼쪽 위인 (0, 0)에서 출발하면 왼쪽 그림과 가운데 그림의 예와 같이, 어떻게 이동하더라도 모든 갈 수 있는 칸을 적어도 한 번 파란색, 그리고 적어도 한 번 노란색으로 칠할 수 없다. 반면, (0, 1)에서 출발하여 오른쪽 그림처럼 이동하면 모든 갈 수 있는 칸을 적어도 한 번 파란색, 그리고 적어도 한 번 노란색으로 칠할 수 있음을 알 수 있다.



격자의 크기와 배치가 주어졌을 때, 모든 갈 수 있는 칸을 파란색과 노란색으로 칠할 수 있는지를 구하려 한다. 여러분은 이 문제를 풀기 위해서 다음 함수를 구현해야만 한다.

- `int yellowblue(int N, int M, vector<string> V)` ; 단 한번 호출되는 함수이다. N 과 M 은 격자의 크기를 나타낸다. V 는 격자의 상태를 나타내는 크기 N 인 string의 배열(vector)이다. V 의 각 string의 길이는 M 이다. 격자의 (i, j) 번째 칸이 갈 수 있는 칸이라면 $V[i]$ 의 j 번째 글자는 '.'이고, 막혀 있는 칸이면 '#'이다. 만약 어떤 위치에서 시작하여 격자의 모든 갈 수 있는 칸을 적어도 한 번 파란색, 적어도 한 번 노란색으로 칠할 수 있다면 1을, 그렇지 않다면 0을 return해야 한다.

구현 세부사항

여러분은 `grid.cpp`라는 이름을 가진 하나의 파일을 제출해야만 한다. 이 파일에는 다음의 함수가 구현되어 있어야 한다.

```
■ int yellowblue( int N, int M, vector<string> V ) ;
```

이 함수는 위에서 설명한 것과 같이 동작하여야 한다. 물론, 다른 함수들을 만들어서 내부적으로 사용할 수 있다. 제출한 코드는 입출력을 수행하거나 다른 파일에 접근하여서는 안된다.

grader 예시

주어지는 grader는 다음과 같은 형식으로 입력을 읽는다.

- line 1: $N M$
(N : 격자의 행의 수, M : 격자의 열의 수)
- line $2+i$ ($0 \leq i \leq N-1$): '.'과 '#'으로 이루어진 길이 M 인 문자열로, 이 문자열의 j 번째($1 \leq j \leq M$) 글자가 '.'이라면 격자의 $(i, j-1)$ 번째 칸은 갈 수 있는 칸이고, '#'이면 격자의 $(i, j-1)$ 번째 칸은 막혀 있는 칸이다.

주어지는 grader는 여러분의 코드가 `yellowblue()` 함수에서 리턴한 값을 출력한다.

제한 조건

- $1 \leq N, M \leq 1,000$, 전체 격자에 갈 수 있는 칸이 적어도 하나 존재한다.

서브태스크 1 [30 points]

- $N, M \leq 50$

서브태스크 2 [70 points]

- 추가 제한이 없다.

[입력 예 1]

```
1 1
.
```

[출력 예 1]

```
1
```

[입력 예 2]

```
2 3
...
...
```

[출력 예 2]

1

[입력 예 3]

```
3 3
...
..#
.##
```

[출력 예 3]

1

[입력 예 4]

```
3 3
.##
#..
#..
```

[출력 예 4]

0