

```
...
```

```
Time complexity:  $O(N*M)$   
Space complexity:  $O(N*M)$   
where N and M are the dimensions of the board
```

```
...
```

```
from sys import stdin  
dx = [1, -1, 0, 0]  
dy = [0, 0, 1, -1]
```

```
def dfs(board, x, y, fromX, fromY, needColor, n, m, visited) :  
    ans = False  
  
    if (x < 0 or x >= n or y < 0 or y >= m) :  
        return ans  
  
    if (board[x][y] != needColor) :  
        return ans  
  
    if (visited[x][y]) :  
        return True  
  
    visited[x][y] = 1  
  
    for f in range(4) :  
        nextX = x + dx[f]  
        nextY = y + dy[f]  
  
        if (nextX == fromX and nextY == fromY) :  
            continue  
  
        ans |= dfs(board, nextX, nextY, x, y, needColor, n, m, visited)  
  
    return ans  
  
def solve(board, n, m) :  
    visited = [[0 for i in range(m + 1)] for j in range(n + 1)]  
    ans = False  
  
    for i in range(n) :
```

```

        for j in range(m) :
            if (not(visited[i][j])) :
                ans |= dfs(board, i, j, -1, -1, board[i][j], n, m, visited)

    return ans

def takeInput():
    #To take fast I/O
    n,m=list(map(int,stdin.readline().strip().split( )))
    arr = [stdin.readline().strip() for i in range(n)]
    return arr,n,m

# Main
arr,n,m = takeInput()
ans = solve(arr,n,m)

if(ans) :
    print('true')
else :
    print('false')

```