```cpp
//
//   Number of paths through a labyrinth.
//
//   Time Complexity: O(NM) = O(size of the matrix)
//

#include <bits/stdc++.h>

using namespace std;

const int MOD = 1e9 + 7;

vector<vector<bool>> board;
vector<vector<int>> dp;

int main() {
    int n, m;
    cin >> n >> m;

    board.resize(n, vector<bool>(m, true));
    dp.resize(n, vector<int>(m, 0));

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            char c; cin >> c;
            board[i][j] = (c == '.');
        }
    }

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            if (i == 0 && j == 0) dp[i][j] = 1;
            else if (i == 0) dp[i][j] = dp[i][j-1] * board[i][j];
            else if (j == 0) dp[i][j] = dp[i-1][j] * board[i][j];
            else {
                dp[i][j] = (dp[i-1][j] + dp[i][j-1]) * board[i][j] % MOD;
            }
        }
    }

    cout << dp[n-1][m-1] << endl;
    return 0;
}
```