using System;

using System.Linq;

using System.Collections.Generic;

namespace Re\_Volt

{

class Program

{

static void Main(string[] args)

{

int size = int.Parse(Console.ReadLine());

int commands = int.Parse(Console.ReadLine());

char[,] matrix = new char[size, size];

int playerRow = 0;

int playerCol = 0;

for (int row = 0; row < size; row++)

{

string currentRow = Console.ReadLine();

for (int col = 0; col < size; col++)

{

if (currentRow[col] == 'f')

{

playerRow = row;

playerCol = col;

}

matrix[row, col] = currentRow[col];

}

}

bool hasWon = false;

for (int i = 0; i < commands; i++)

{

var command = Console.ReadLine();

if (command == "right")

{

setPlayerPosToDash(playerRow, playerCol, matrix);

playerCol++;

if (isOutside(playerRow, playerCol, size))

{

playerCol = 0;

}

if (checkIfBonus(playerRow, playerCol, matrix))

{

playerCol++;

if (isOutside(playerRow, playerCol, size))

{

playerCol = 0;

}

}

else if (checkIfTrap(playerRow, playerCol, matrix))

{

playerCol--;

if (isOutside(playerRow, playerCol, size))

{

playerCol = size - 1;

}

}

if (checkIfGoal(playerRow, playerCol, matrix))

{

hasWon = true;

setNewPlayerPos(playerRow, playerCol, matrix);

goto End;

}

setNewPlayerPos(playerRow, playerCol, matrix);

}

else if (command == "left")

{

setPlayerPosToDash(playerRow, playerCol, matrix);

playerCol--;

if (isOutside(playerRow, playerCol, size))

{

playerCol = size - 1;

}

if (checkIfBonus(playerRow, playerCol, matrix))

{

playerCol--;

if (isOutside(playerRow, playerCol, size))

{

playerCol = size - 1;

}

}

else if (checkIfTrap(playerRow, playerCol, matrix))

{

playerCol++;

if (isOutside(playerRow, playerCol, size))

{

playerCol = 0;

}

}

if (checkIfGoal(playerRow, playerCol, matrix))

{

hasWon = true;

setNewPlayerPos(playerRow, playerCol, matrix);

goto End;

}

setNewPlayerPos(playerRow, playerCol, matrix);

}

else if (command == "down")

{

setPlayerPosToDash(playerRow, playerCol, matrix);

playerRow++;

if (isOutside(playerRow, playerCol, size))

{

playerRow = 0;

}

if (checkIfBonus(playerRow, playerCol, matrix))

{

playerRow++;

if (isOutside(playerRow, playerCol, size))

{

playerRow = 0;

}

}

else if (checkIfTrap(playerRow, playerCol, matrix))

{

playerRow--;

if (isOutside(playerRow, playerCol, size))

{

playerRow = size - 1;

}

}

if (checkIfGoal(playerRow, playerCol, matrix))

{

hasWon = true;

setNewPlayerPos(playerRow, playerCol, matrix);

goto End;

}

setNewPlayerPos(playerRow, playerCol, matrix);

}

else if (command == "up")

{

setPlayerPosToDash(playerRow, playerCol, matrix);

playerRow--;

if (isOutside(playerRow, playerCol, size))

{

playerRow = size - 1;

}

if (checkIfBonus(playerRow, playerCol, matrix))

{

playerRow--;

if (isOutside(playerRow, playerCol, size))

{

playerRow = size - 1;

}

}

else if (checkIfTrap(playerRow, playerCol, matrix))

{

playerRow++;

if (isOutside(playerRow, playerCol, size))

{

playerRow = 0;

}

}

if (checkIfGoal(playerRow, playerCol, matrix))

{

hasWon = true;

setNewPlayerPos(playerRow, playerCol, matrix);

goto End;

}

setNewPlayerPos(playerRow, playerCol, matrix);

}

}

End:;

if (hasWon)

{

Console.WriteLine($"Player won!");

}

else

{

Console.WriteLine($"Player lost!");

}

for (int row = 0; row < size; row++)

{

for (int col = 0; col < size; col++)

{

Console.Write(matrix[row,col]);

}

Console.WriteLine();

}

}

public static bool isOutside(int row, int col, int size)

{

if (row >= size || col >= size || row < 0 || col < 0)

{

return true;

}

return false;

}

public static void setPlayerPosToDash(int playerRow, int playerCol, char[,] matrix)

{

matrix[playerRow, playerCol] = '-';

}

public static void setNewPlayerPos(int playerRow, int playerCol, char[,] matrix)

{

matrix[playerRow, playerCol] = 'f';

}

public static bool checkIfTrap(int playerRow, int PlayerCol, char[,] matrix)

{

if (matrix[playerRow, PlayerCol] == 'T')

{

return true;

}

return false;

}

public static bool checkIfBonus(int playerRow, int PlayerCol, char[,] matrix)

{

if (matrix[playerRow, PlayerCol] == 'B')

{

return true;

}

return false;

}

public static bool checkIfGoal(int playerRow, int PlayerCol, char[,] matrix)

{

if (matrix[playerRow, PlayerCol] == 'F')

{

return true;

}

return false;

}

}

}