**Matrix Symmetry**

Attempted by: **805**

/

Accuracy: **71%**

/

Maximum Score: **20**

/

7 Votes

Tag(s):

Ad-Hoc, Easy, Implementation

**PROBLEM**

**EDITORIAL**

**MY SUBMISSIONS**

**ANALYTICS**

You are given a square matrix of size nn. Rows are indexed 11 to nn from top to bottom and columns are indexed 11 to nn form left to right. Matrix consists of only '\*' and '.'. You need to check whether matrix is symmetric or not. if it is, check it is symmetric about vertical axis or horizontal axis or both.

A matrix is said to be symmetric about horizontal axis if 1st1st row is identical to nthnth row, 2nd2nd is identical to (n−1)th(n−1)th row and so on...

A matrix is said to be symmetric about vertical axis if 1st1st column is identical to nth column, 2nd2nd identical to (n−1)th(n−1)th and so on for all columns.

**INPUT** :

First line contains tt,the number of test cases. First line of each test case contains nn the size of matrix. Each of next nn lines contain nn characters.

**OUTPUT**:

Output t lines, answer for each test case. Print "HORIZONTAL" if symmetric about horizontal axis. Print "VERTICAL" if symmetric about vertical axis. Print "BOTH" if symmetric about both axes. print "NO" if it is not symmetric.

**Constraints** :

1<t≤5001<t≤500   
1<n<501<n<50

**SAMPLE INPUT**

3

4

\*.\*.

.\*.\*

\*.\*.

.\*.\*

3

.\*.

\*.\*

.\*.

3

..\*

\*\*.

..\*

**SAMPLE OUTPUT**

NO

BOTH

HORIZONTAL

**Time Limit:**2.0 sec(s) for each input file.

**Memory Limit:**256 MB

**Source Limit:**1024 KB

**Marking Scheme:**Marks are awarded when all the testcases pass.

**Allowed Languages:**C, C++, Clojure, C#, D, Erlang, F#, Go, Groovy, Haskell, Java, Java 8, JavaScript(Rhino), JavaScript(Node.js), Lisp, Lisp (SBCL), Lua, Objective-C, OCaml, Octave, Pascal, Perl, PHP, Python, Python 3, R(RScript), Racket, Ruby, Rust, Scala, Scala 2.11.8, Swift, Visual Basic

<https://www.hackerearth.com/practice/basic-programming/implementation/basics-of-implementation/practice-problems/algorithm/matrix-symmetry/>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

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

while (t-- > 0)

{

//string[] matrix =

//{

// "\*.\*.",

// ".\*.\*",

// "\*.\*.",

// ".\*.\*"

//};

//string[] matrix =

//{

// ".\*.",

// "\*.\*",

// ".\*."

//};

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

string[] matrix = new string[n];

int i,j;

for (i = 0; i < n; i++)

{

matrix[i] = Console.ReadLine();

}

i = 0;

j = matrix[0].Length-1;

bool vertical = true;

while (i < j && vertical)

{

for (int fila = 0; fila < matrix.Length; fila++)

{

if (matrix[fila][i] != matrix[fila][j])

{

vertical = false;

break;

}

}

i++;

j--;

}

i = 0;

j = matrix.Length-1;

bool horizontal = true;

while (i < j && horizontal)

{

for (int col = 0; col < matrix[0].Length; col++)

{

if (matrix[i][col] != matrix[j][col])

{

horizontal = false;

break;

}

}

i++;

j--;

}

if (horizontal && vertical)

{

Console.WriteLine("BOTH");

}

else if (horizontal)

{

Console.WriteLine("HORIZONTAL");

}

else if (vertical)

{

Console.WriteLine("VERTICAL");

}

else

{

Console.WriteLine("NO");

}

}

Console.ReadLine();

}

}

}