**Roy and Symmetric Logos**

Attempted by: **599**

/

Accuracy: **81%**

/

Maximum Score: **20**

/

5 Votes

Tag(s):

Ad-Hoc, Basic Programming, Easy, Geometry, Implementation

**PROBLEM**

**EDITORIAL**

**MY SUBMISSIONS**

**ANALYTICS**

Roy likes Symmetric Logos.

*How to check whether a logo is symmetric?*  
Align the center of logo with the origin of Cartesian plane. Now if the colored pixels of the logo are symmetric about both X-axis and Y-axis, then the logo is symmetric.

You are given a binary matrix of size **N x N** which represents the pixels of a logo.  
**1** indicates that the pixel is colored and **0** indicates no color.

For instance: Take a 5x5 matrix as follows:

01110

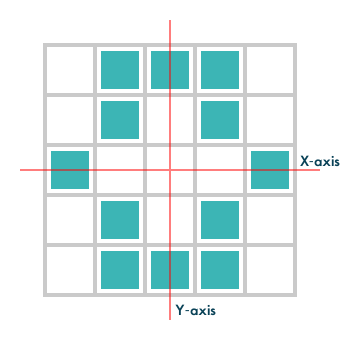
01010

10001

01010

01110

Graphically it is represented as follows:



Observe that it is symmetric about both X-axis and Y-axis.

Let's take another example of 5x5 matrix:

00100

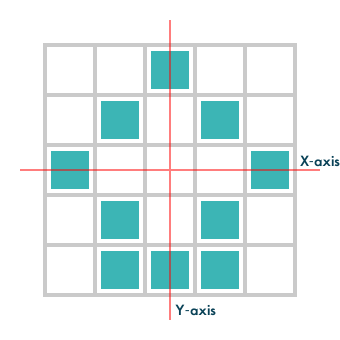
01010

10001

01010

01110

Graphically it is represented as follows:



Now this logo is symmetric about Y-axis but it is **not symmetric** about X-axis.

Your task is to output **YES** if the logo is symmetric else output **NO**.

**Input:**  
First line contains **T** - number of test cases.  
T test cases follow.  
First line of each test case contains the **N** - size of matrix.  
Next N lines contains **binary strings** of length **N**.

**Output:**  
Print **YES** or **NO** in a new line for each test case

**Constraints:**   
**1 ≤ T ≤ 10  
2 ≤ N ≤ 32**

**Note:** There will always be atleast 1 colored pixel in input data.

**SAMPLE INPUT**

5

2

11

11

4

0101

0110

0110

0101

4

1001

0000

0000

1001

5

01110

01010

10001

01010

01110

5

00100

01010

10001

01010

01110

**SAMPLE OUTPUT**

YES

NO

YES

YES

NO

**Explanation**

Test Case #1: Symmetric about both axes, so YES.

Test Case #2: Symmetric about X-axis but not symmetric about Y-axis, so NO.

Test Case #3: Symmetric about both axes, so YES.

Test Case #4 and #5 are explained in statement.

**Time Limit:**5.0 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/data-structures/arrays/multi-dimensional/practice-problems/algorithm/roy-and-symmetric-logos-1/>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static string esSimetrica(string[] m)

{

int i = 0, j = m[0].Length - 1;

while (i < j)

{

for (int k = 0; k < m.Length; k++)

{

if (m[i][k] != m[j][k])

{

return "NO";

}

}

i++;

j--;

}

//i = 0;

//j = m.Length - 1;

for (int k = 0; k < m.Length; k++)

{

i = 0; j = m[k].Length - 1;

while (i < j)

{

if (m[k][i] != m[k][j])

{

return "NO";

}

i++;

j--;

}

}

return "YES";

}

static void Main(string[] args)

{

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

while (t-- > 0)

{

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

string[] matrix = new string[n];

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

{

matrix[i] = Console.ReadLine();

}

Console.WriteLine(esSimetrica(matrix));

}

Console.ReadLine();

}

}

}