

# E1. Photographs (I)

time limit per test: 15 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

The Human-Cow Confederation ( $HC^2$ ), led by Heidi, has built a base where people and cows can hide, guarded from zombie attacks. The entrance to the base is protected by an automated gate which performs a kind of a Turing test: it shows the entering creature a photograph and asks them whether the top and bottom halves of this photograph have been swapped or not. A person (or even a cow) will have no problem answering such questions; on the other hand, a zombie would just randomly smash one of the two buttons.

The creature is asked a series of such questions. If at least 75% of them are answered correctly, the gate is unlocked; otherwise, a side door opens, beneath which a huge fan is spinning...

Heidi is now building a robot army to fight the zombies, and she wants the robots to also be able to enter the base. You are tasked with programming them to distinguish the images.

The first two images from the test set. The first picture has been rearranged, but not the second.

## Input

The first line of the input contains the number  $q$  of questions ( $1 \leq q \leq 220$ ). After that,  $q$  questions follow, each of which in the format described below.

The first line of every question contains two space-separated integers  $h$  and  $w$  ( $1 \leq h, w \leq 600$ ) – the height (number of rows) and width (number of columns) of the photograph. (Most photographs are roughly  $200 \times 300$ .) After this,  $h$  lines follow, each describing a single row of the picture. The picture is monochrome (in shades of grey). Its  $i$ -th row is described by  $w$  space-separated integers  $a_{ij}$  ( $j = 1, \dots, w$ ), where  $a_{ij}$  is the brightness of the corresponding pixel ( $0 \leq a_{ij} < 256$ , where 0 is black and 255 is white).

Each picture will be either a real-life photograph, or a real-life photograph which has been broken up into two pieces and rearranged. More precisely, in the latter case, the topmost rows have been moved to the bottom of the picture. It is guaranteed that  $h$  is even.

There is only a single input file to be processed, called `all.in`, and it is downloadable from the online judge. You are also given another input file, called `sample.in`, which contains the first 20 pictures from `all.in`; you are provided the correct answers for `sample.in` in `sample.out`. You are also given a directory `easy_bmp`, which contains the first 50 input photographs in the form of `.bmp` image files, as well as a directory `easy_sample_original_bmp`, which contains the first 20 images *before rearrangement*. Check the notes for the download links.

## Output

Your program should print  $q$  lines. The  $i$ -th line should contain your answer for the  $i$ -th question: YES if the photograph has been rearranged and NO otherwise. Your answers will be accepted if they all conform to this format and if at least 75% of them are correct.

Because the input is rather huge, feel free to process it locally and submit just your precomputed answers (i.e., a program which just prints your output for the input file `all.in`).

## Note

The link to download all necessary files is [http://assets.codeforces.com/files/690/easy\\_contestant\\_package.zip](http://assets.codeforces.com/files/690/easy_contestant_package.zip)