

2858 Blurred Vision

Aliasing is the stair-step effect achieved when attempting to represent a smooth curve using a finite number of discrete pixels. Of course, all computer displays consist of a finite number of pixels, and many strategies have been devised to smooth the jagged edges with varying degrees of success.

Boudreaux and Thibodeaux are writing video game rendering software for the next big first-person shooter, and they don't know much about any of the progress made in the field of anti-aliasing. Therefore, they've decided to use a very simplistic (and visually unappealing) method to smooth the ragged edges. Unfortunately, it blurs the entire image, but at least it gets rid of those jaggies!

Normally, the game displays in $m \times n$ pixels, but they perform an extra anti-aliasing step that converts that image into an $(m - 1) \times (n - 1)$ image. Nobody will notice a pixel missing from each dimension, and they can calculate the new pixels by averaging squares of 4 pixels from the original image (and rounding down).

4	4	4	0
4	4	0	0
4	0	0	0
0	0	0	0

4	3	1
3	1	0
1	0	0

For example, the images above represent the original image (left) and the anti-aliased image (right) using numbers to represent varying shades of black and white.

Input

Input to this problem will consist of a (non-empty) series of up to 100 data sets. Each data set will be formatted according to the following description, and there will be **no blank lines** separating data sets.

A single data set has 3 components:

1. *Start line* — A single line:

START R C

where R and C are integers ($2 \leq (R, C) \leq 9$) indicating the number of rows and columns in the input image described by this data set.

2. *Original Image* — A series of R lines, each of which contains C integers from 0 to 9 inclusive. These integers represent the grayscale value of a pixel in the original image and will not be separated by spaces.

3. *End line* — A single line:

END

After the last data set, there will be a single line:

ENDOFINPUT

Output

The output will be the anti-aliased image, which will be $R - 1$ rows, each with $C - 1$ integer pixel values. Each pixel in the output will be generated by averaging (and rounding down) the grayscale pixel values of the corresponding square of four pixels in the Original Image.

Sample Input

```
START 2 2
00
00
END
START 2 9
012345678
012345678
END
START 4 4
4440
4400
4000
0000
END
START 9 9
900000009
090000090
009000900
000909000
000909000
009000900
090000090
900000009
END
ENDOFINPUT
```

Sample Output

```
0
01234567
431
310
100
42000024
24200242
02422420
00244200
00244200
02422420
24200242
42000024
```