DSA Lab 9 | Compression

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

This lab assignment is related to data compression. You will be required to compress a gray scale image using Huffman coding. A sample image is displayed here.

You will be provided with a pixel matrix containing gray values. The gray value of each pixel of the image can be represented by 8 bits, hence the range of gray value in decimal representation is between 0 and 255. Create a priority queue to hold the Huffman tree and encode the gray values, as discussed in the lecture class.

Part 1

Output the compression ratio of the image rounded off to one decimal place. The **Compression ratio** is the number of bits in the original image divided by the number of bits in the compressed image.

Part 2

The image can be compressed further by ignoring minor variations in gray values. To do this part, work out the histogram of the image (number of pixels corresponding to each gray value). Consider the ranges $0-9$, $10-19$, $20-29$,..., $230-239$, $240-255$. Find the gray value having the highest number of pixels in each range, and assign that gray value to all pixels falling in the range.

Gray Value	30	31	32	33	34	35	36	37	38	39
Number of Pixels	7	216	45	0	126	723	328	100	0	30

Assign the gray value 35 to all pixels having gray value 30-39 in the original image, as the maximum number of pixels have this value. Incase, multiple gray values in a given range have the same frequency, then assign the **smallest** such gray value to all pixels having gray value in the considered range. The resultant image will have fewer gray values and can be coded with fewer bits.

Output the compression ratio of the image for this part (rounded off to one decimal place).

Input

The first line contains two integers x and y (separated by space), representing width and height of the pixel matrix, respectively. Next y lines contain x space separated integers in each line, denoting gray value for pixels in each row of the matrix.

Constraints: Basic:
$1 \leq x, y \leq 20$
$0 \leq matrix_{x,y} \leq 40$
Advanced:
$1 \leq x, y \leq 1000$
$0 \leq matrix_{x,y} \leq 255$

Output

The decimal value (rounded off to one decimal place) representing the compression ratio for Part 1.
The decimal value (rounded off to one decimal place) representing the compression ratio for Part 2.

Example

standard input	standard output
4 4 109 35 36 40 35 36 55 60 40 45 100 105 35 36 40 45	2.6 3.9

Note

Description for the sample test case
As every pixel occupies 8 bits in original image, size before compression is $4 * 4 * 8$ bits $= 128$ bits
Bits occupied after compression in part $1 = 49$ bits
Compression ratio for part $1 = \frac{128}{49} = 2.6$
Bits occupied after compression in part $2 = 33$ bits
Compression ratio for part $2 = \frac{128}{33} = 3.9$