

Problem 95: Image Compression

Difficulty: Easy

Author: Steve Brailsford, Marietta, Georgia, United States

Originally Published: Code Quest 2019

Problem Background

Images can be saved onto a computer in many different types of file formats, each with its own advantages and disadvantages. JPEG (or JPG) images are commonly used for photography, because their format allows the image information to be compressed, reducing the size of the file and allowing you to take more pictures. The downside to this is that repeatedly editing a JPEG image causes the quality of the image to gradually get worse over time; each time the file is saved, the existing image data is compressed further and further, losing fine details.

The process of compressing a JPEG image is complicated but can be broken down into several individual steps. One of these steps is called quantization, which takes a wide range of numbers created by a previous step in the process and converts them to a smaller, more manageable scale. This results in some loss of detail as previously mentioned; two different but close numbers may be converted to the same result number. However, the human eye often cannot discern very high-frequency changes, so this loss is usually not noticeable.



Problem Description

Your program will need to implement an example quantization algorithm that accepts perceived brightness values and converts them to an integer value between 0 and 255 inclusive. Your program will be given a list of decimal values representing brightness values (such as might be read by a scanner). Your program must identify the highest (max) value and the lowest (min) value from the list of values, then convert all values in the list to the target scale using this formula:

$$Output = \frac{Input - Min}{Max - Min} * 255$$

All results should be rounded to the nearest integer.

Sample Input

The first line of your program's input, received from the standard input channel, will contain a positive integer representing the number of test cases. Each test case will include the following lines of input:

- A positive integer, X , representing the number of values in the list
- X lines, each containing a decimal number to be converted

```
2
5
0.0
25.0
50.0
75.0
100.0
6
12.3
-67.1
122.8
428.4
-15.9
221.0
```

Sample Output

For each test case, your program must output the list of converted numbers, maintaining the same order. Print one number per line, and round all results to the nearest integer.

```
0
64
128
191
255
41
0
98
255
26
148
```

The image for this problem is licensed under the Creative Commons Attribution 3.0 Unported License (see <https://creativecommons.org/licenses/by/3.0/> for more info) and was created by Michael Gäbler and AzaToth from the Wikimedia Commons. It is free for use provided that attribution is given to the authors along with the terms of this license.