**Problem #1**

**Run-length Encoding (RLE)**

Run-length encoding (RLE) is a very simple form of data compression in which runs of data (that is, sequences in which the same data value occurs in many consecutive data elements) are stored as a single data value and count, rather than as the original run. This is most useful on data that contains many such runs. Consider, for example, a black and white image, where each pixel can only have a value of 0 (black) or 1 (white).

See the following 5X6 image data. Notice how it is converted in RLE. Firstly the dimension of the image is printed out. After a space, the actual data is printed in RLE format. Each run of same color is represented with the length of the run, followed by its color. A space separator is used between each such run. Carefully notice that, when there is no run (i.e. a white is immediately followed by black or vice versa), the colors are directly mentioned instead:

|  |  |
| --- | --- |
| **Original Image Data** | **RLE Output** |
| 0 0 1 1 1 1  1 1 0 0 0 1  1 0 0 1 0 1  0 0 0 1 1 0  1 0 1 0 0 0 | 5X6 2B 6W 3B 2W 2B W B W 3B 2W B W B W 3B |

The first line of input contains *m* and *n,* representing respectively the height and width of the image in pixels. (1 <= *m, n* <= 10). Then *m* lines follow, each containing *n* space separated integers from {0, 1}, representing the pixel colors of the image.

You need to output the run-length encoded image.

|  |  |
| --- | --- |
| **Sample Input(s)** | **Corresponding Output(s)** |
| 5 6  0 0 1 1 1 1  1 1 0 0 0 1  1 0 0 1 0 1  0 0 0 1 1 0  1 0 1 0 0 0 | 5X6 2B 6W 3B 2W 2B W B W 3B 2W B W B W 3B |
| 4 7  0 0 1 1 1 1 1  1 1 0 0 0 1 0  1 0 0 1 0 1 1  0 0 0 1 1 1 0 | 4X7 2B 7W 3B W B W 2B W B 2W 3B 3W B |