Problem I
2D Max Filter

ACM-ICPC Thailand
Central Group B
Programming Contest 2013

An electrical engineer wants to implement an efficient algorithm for filtering a 2D signal that is digitized into blocks with *R* rows and *C* columns. Each block is associated with its value digitized to a positive integer. Figure 1 shows example values of a 2D signal with 5 rows and 8 columns.

```
0 7 8 9 5 3 2 7
1 5 2 3 4 2 8 7
1 0 2 2 2 1 6 8
4 2 3 1 0 7 2 1
5 2 1 0 4 3 0 7
```

Figure 1. Example values of an input signal.

Regarding the filter, it has a rectangular window covering a region of (2*M* + 1) rows x (2*N* + 1) columns of a signal, where *M* and *N* are non-negative integers. The window moves through the entire 2D signal and produces another 2D signal under the following rules:

1. When a window center is at row A and column B of an input signal, the window will produce a value in an output signal at row A and column B.
2. The produced value is the maximum input value inside a region covered by the window.
3. A part of a window that is outside the input signal is not considered. Only a part of a window inside an input signal is taken into account.

According to the rules, if *M* is 1 and *N* is 2, the window size is 3 rows x 5 columns and the output signal is displayed in Figure 2. (See explanation of how this is calculated below.)

```
8 9 9 9 9 9 8 8
8 9 9 9 9 9 8 8
5 5 5 7 8 8 8 8
5 5 5 7 7 8 8 8
5 5 5 7 7 7 7 7
```

Figure 2. Output values produced from the signal in Figure 1 when a filter's window size is 3 rows x 5 columns.

For the sake of concreteness, let's focus on the output at row 4, column 5 (the row and column indices start from 1). This output value is produced when the window center is also at row 4, column 5. Therefore, the window covers the input region from row 3, column 3 to row 5, column 7 (highlighted in Figure 3). We can see that the maximum value inside the window region is 7 and this becomes the output produced from the window at row 4, column 5.

```
0 7 8 9 5 3 2 7
1 5 2 3 4 2 8 7
1 0 2 2 2 1 6 8
4 2 3 1 0 7 2 1
5 2 1 0 4 3 0 7
```

Figure 3. Window coverage when it is centered at row 4, column 5.

Now, let's focus on the output at row 1, column 2. Technically, the window covers the input region from row -1, column -1 to row 2, column 4. However, since the region outside the input signal is ignored, the input values taken into account are those from row 1, column 1 to row 2 column 4 (highlighted in Figure 4). Consequently, the maximum value is 9 and it is the output value for row 1, column 2.

0 7 8 9 5 3 2 7
1 5 2 3 4 2 8 7
1 0 2 2 2 1 6 8
4 2 3 1 0 7 2 1
5 2 1 0 4 3 0 7

Figure 4. Window coverage when it is centered at row 1, column 2.

Your task is to write an efficient algorithm for this 2D max filter for the input whose specification is shown next.

Input

1. The first line contains R and C, the numbers of rows and columns of an input signal, respectively, where 1 <= R <= 1000 and 1 <= C <= 1000. These values are separated by a space.
2. The second line contains M and N, respectively, where 0 <= M <= 60 and 0 <= N <= 60. A window size is calculated from M and N by (2M + 1) rows x (2N + 1) columns. Values M and N are also separated by a space.
3. Line 3 contains C integers separated by a space. These are values in the first row of an input signal from left to right. Each value is a positive integer ranging from 1 to 10000, including.
4. Lines 4 to R + 2 are similar to Line 3, but contain input signal values from rows 2 to *R*.

Output

1. The first line contains C integers which are output values in the first row of the output signal. Values are separated by a space.
2. Lines 2 to *R* are similar to the first line, but contain output signal values from rows 2 to *R*.

Note: the end of each line may contain one trailing white space and there can be one trailing empty line at the end of output.

Example

Input	**Output**
5 8 1 2 0 7 8 9 5 3 2 7 1 5 2 3 4 2 8 7 1 0 2 2 2 1 6 8 4 2 3 1 0 7 2 1 5 2 1 0 4 3 0 7	8 9 9 9 9 9 8 8 8 9 9 9 9 9 8 8 5 5 5 7 8 8 8 8 5 5 5 7 7 8 8 8 5 5 5 7 7 7 7 7