Many problems in image processing and computer graphics need to process large two dimensional arrays and with modern real time applications they need to process them very **fast**. For instance when one wishes to blur an image a “Convolution” filter is applied, this involves nothing more than summing the values near or around a particular pixel and applying some mathematical function for instance calculating the average value. More complicated functions exist, the Gaussian Blur being quite popular. In a Gaussian Blur when summing not all pixel values have equal weights in the sum, in other words pixel weights aren't equal - they are given a value according to a bell-shaped Gaussian curve.

This problem involves computing local information, very similar to “blurring” an image, but a shade simpler! You will be given an $N \times N$ array of integer data and your task will be to calculate an $(N - M + 1) \times (N - M + 1)$ array of integers. Each element of this resulting matrix is the sum of the local $M \times M$ points of the original matrix.

For instance, consider the following example for a 4 by 4 Matrix X with $M = 3$.

$$\text{For } X = \begin{pmatrix} 1 & 2 & 1 & 3 \\ 0 & 2 & 3 & 3 \\ 1 & 0 & 2 & 2 \\ 1 & 2 & 2 & 1 \end{pmatrix} \text{ the resultant “blurred” matrix is } \begin{pmatrix} 12 & 18 \\ 13 & 17 \end{pmatrix}$$

The highlighted matrix elements show the points to be summed at the first element of the matrix. The first element of a matrix is the bottom left hand corner (in this diagram).

Input

The input will contain several test cases, each of them as described below. Consecutive test cases are separated by a single blank line.

The input values consist of $(N \times N) + 1$ lines.

The first line contains the integer values of N and M separated by a single space.

The next $N \times N$ lines contain the elements of the matrix, one per line, from the bottom left element to the top right one.

The maximum value of N is 1000. M is always greater or equal to 2 and less than N.

Each element of the matrix is a positive integer value less than 16 (0,1,..15).

Output

For each test case, the output must follow the description below. The outputs of two consecutive cases will be separated by a blank line.

The output data consists of the elements of the resulting matrix plus one extra integer value, the total sum of the elements.

The elements of the “blurred” matrix are displayed one per line, starting from the bottom left to the top right element.

The last line contains the total sum. This sum is guaranteed to be at most a 64 bit integer.

Note: The first sample input represents the input and output data for the example described above.

Sample Input

```
4 3
1
2
2
1
1
0
2
2
0
2
3
3
1
2
1
3

5 2
1
0
1
0
0
0
0
1
0
0
1
1
1
1
1
0
0
1
0
0
0
0
1
0
0
```

Sample Output

```
13
17
12
18
60

1
2
2
0
2
3
3
2
2
3
3
2
0
2
2
0
29
```