Lab 11 Tasks

Q1. The slogan of the recent fresher's reception got of the CSE department made some of the students curious about computer displays and how images are represented in computers. The students found out that when any picture is stored in a computer the picture is divided into a ***n x n*** grid where *n* is a positive integer. Each cell of the grid contains an integer value which represents a specific shade of a color. What they found even more interesting is that they could now easily implement some of the simple filters that they use in Snapchat, Facebook and Instagram. Your task is to implement one of these filters which makes an image look smoother.

You already know an image will be represented by a *n x n* integer array where *n* is an integer. A neighbor distance, *d* is also specified where *d* is an integer. For any given index (i, j) in the array, all the indices that are within distance *d* are considered to be the neighbors of index (i, j). For example, in the figure below, if *d* = 1 then the neighbors of the index (1,1) are (0,0), (0,1), (0,2), (1,0), (1,2), (2,0), (2,1), (2,2). For *d* = 2, all the indices of the array would be the neighbor of (2,2).

5 (0,0)	7 (0,1)	5 (0,2)	4 (0,3)	0 (0,4)
8 (1,0)	1 (1,1)	15 (1,2)	2 (1,3)	11 (1,4)
3 (2,0)	4 (2,1)	2 (2,2)	13 (2,3)	4 (2,4)
9 (3,0)	12 (3,1)	7 (3,2)	8 (3,3)	6 (3,4)
11 (4,0)	3 (4,1)	8 (4,2)	15 (4,3)	12 (4,4)

Now to make the image smooth, follow these steps:

1. Take the first index (0, 0)
2. Find the average of the values stored in the neighbor indices and also the current index.
3. Store this value in the current index location. If the index of any neighbor is out of bound then ignore that neighbor. For example, the value to be stored in index (0, 0) for *d* = 1 will be $\frac{7+8+1}{3}$
4. Repeat steps 2 and 3 for every index in the array.

Input

The first line of input will be the integers *n* and *d.* The following lines will be the *n* x *n* array.

Output

The output will be the *n* x *n* after applying the filter.

Example

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

Q2. Given an integer *n,* the numbers [1, 2, 3, …., n] can be rearranged to create a total of *n*! different numbers each with *n* digits. For example, if *n* = 4, the following numbers can be created.

1. 1234	7. 2134	13. 3124	19. 4123
2. 1243	8. 2143	14. 3142	20. 4132
3. 1324	9. 2314	15. 3214	21. 4213
4. 1342	10. 2341	16. 3241	22. 4231
5. 1423	11. 2413	17. 3412	23. 4312
6. 1432	12. 2431	18. 3421	24. 4321

Here all the numbers are arranged in ascending order. So from this list if I wanted to find the *s* = 18th number it would be 3421.

Given the values of *n* and *d,* find the d^{th} number from the sorted list containing all the numbers that can be created using the [1, 2, …, n] digits.

Example

Input	**Output**
4 11	2413

Q3. Given an integer *n*, determine if *n* is a power of 3.

Example

Input	**Output**
15	False
Input	**Output**
27	True