

## Combination

**Input File: CombinationIn.txt**

Luck Smith is a locksmith who needs your help with his new design for a digital lock. Each number in a lock's combination is entered into its own cell of an  $n \times n$  square grid of numbers (left side of the figure below for  $n = 3$ ), with the range of the numbers being 0 to  $n - 1$  inclusive. Adjacent to each cell around the perimeter of the lock is a number also in the range 0 to  $n - 1$  inclusive, as shown on the center of the figure for  $n = 3$ . The lock will open when a number is placed in each cell such that the sum of it and the numbers immediately above, below, left and right of it Mod  $n$  is zero, as shown on the right side of the figure.

0 0 0	0 1 1 1 0	0 1 1 1 0
0 0 0	1 0 0 0 2	1 1 0 1 2
0 0 0	1 0 0 0 2	1 0 0 2 2
	1 0 0 0 2	1 1 1 1 2
	0 0 0 0 0	0 0 0 0 0

**Number Positions  
for a 3 x 3 Lock**

**A 3 x 3 Lock**

**A Combination  
for the Lock**

Your task is to determine if there is a valid combination for each of Luck Smith's new locks, and determine a combination that will open the lock for each lock that has a combination. Note: there may be more than one combination for a lock.

### Inputs

The first line of input contains the number of locks Luck Smith has produced. For each of these locks there will be five lines of input. The first line of input will be the number of cells in each of the lock's rows and columns. This will be followed by four lines containing  $n$  numbers that are the perimeter values adjacent to the top, bottom, left, and right sides of the lock's cells respectively. The ordering of these perimeter values on each input line are as shown in the second set of inputs/outputs given below.

### Outputs

For locks that do not have a combination, there will be one line of output containing the word false. For locks that do have a combination, the first line of output will contain the word true, followed by  $n + 2$  lines of output that show a valid combination for the lock and the perimeter values formatted exactly as shown on the right side of the figure above (with one space between each of the numbers on a line). There will be a blank line separating each lock's output.

(sample inputs and outputs are on the next page)

### Sample Inputs

```
4
3
1 1 1
0 0 0
1 1 1
2 2 2
6
1 2 3 1 5 1
1 1 2 1 2 4
2 1 2 1 4 1
3 2 3 2 5 5
4
1 1 1 1
0 1 1 0
1 1 1 1
2 2 0 2
2
1 1
0 0
1 1
0 0
```

### Sample Outputs

```
true
0 1 1 1 0
1 1 0 1 2
1 0 0 2 2
1 1 1 1 2
0 0 0 0 0

true
0 1 2 3 1 5 1 0
2 4 1 1 3 3 4 3
1 4 4 4 4 3 1 2
2 5 5 5 4 1 2 3
1 2 5 0 4 2 5 2
4 5 0 4 2 0 1 5
1 1 4 0 2 1 1 5
0 1 1 2 1 2 4 0

false

true
0 1 1 0
1 0 1 0
1 1 0 0
0 0 0 0
```