J • Pinemi Puzzles

A *Pinemi Puzzle* is a 10-by-10 array of squares with numbers in some of the squares and the other squares left blank.

| **Input** | **Solution** |

To solve the puzzle, add 1, 2, or 3 vertical strokes to each blank box so that:
1. There are exactly 10 strokes in each row
2. There are exactly 10 strokes in each column
3. For each box with a number in it, the number in the box is the total number of strokes in neighboring boxes (up, down, right, left, and diagonal).

For this problem, you will write a program to solve *Pinemi Puzzles*!

Input

The first line of input contains a single decimal integer P, ($1 \leq P \leq 100$), which is the number of data sets that follow. Each data set should be processed identically and independently.

Each data set consists of 11 lines of input. The first line contains the data set number, K. The following 10 lines each consist of 10 integers separated by spaces. A value of -1 indicates that the box in the puzzle is blank. Otherwise, the value gives the count of neighboring strokes for the box. The ith element of the jth of the 10 input lines describes the box in row j and column i.

Output

For each data set there are 11 lines of output.

The first output line consists of the data set number, K. The following 10 lines of output consist of 10 decimal digits separated by one to six spaces. The value in the jth position in the ith line of the 10 output lines is the input value, if the input value in in the jth position in the ith line was > 0. Otherwise it is the number of strokes for the (initially blank) box in the jth column in the ith row. Note: For some problems, there may be more than one correct solution. The number of spaces between values is not important, however, padding the numbers out (as shown below in the **Sample Output**) may make your output more readable and easier to debug.

Sample Input	Sample Output
1	1
1	
-1 9 -1 -1 5 4 -1 5 -1 3	3 9 1 2 5 4 2 5 2 3
5 -1 -1 -1 8 -1 10 -1 7 -1	5 2 3 1 8 2 10 1 7 1
2 10 -1 14 -1 -1 -1 -1 9 -1	2 10 2 14 2 1 2 2 9 1
1 6 -1 -1 -1 12 10 9 -1 -1	1 6 3 2 1 12 10 9 1 3
-1 8 8 -1 -1 -1 -1 7 -1 -1	1 8 8 1 1 2 3 7 1 1
-1 -1 6 6 -1 -1 -1 -1 7 4	2 2 6 6 3 1 1 1 7 4
10 -1 11 -1 9 -1 -1 -1 -1 3	10 2 11 1 9 2 1 2 2 3
-1 -1 -1 -1 7 5 9 -1 -1 7	1 3 1 2 7 5 9 2 1 7
-1 9 8 7 -1 6 7 -1 -1 -1	2 9 8 7 2 6 7 2 2 2
-1 -1 2 -1 -1 -1 -1 6 -1 -1	1 1 2 1 1 2 1 6 1 2

Note: The sample input and output correspond to the pictures in the problem descriptions.