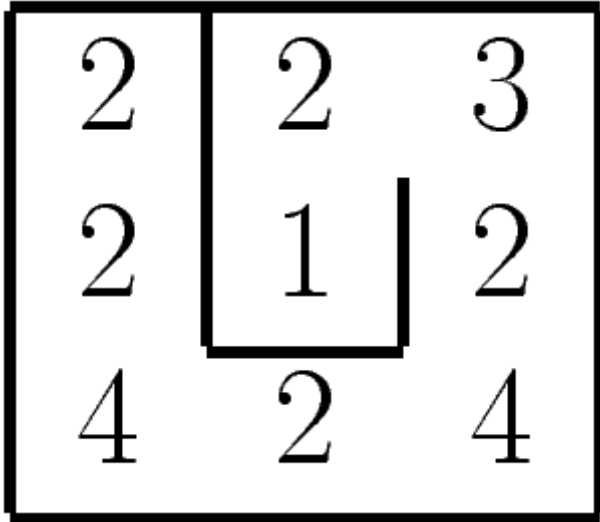


Usually, labyrinths are to be solved by finding a path from entry to exit. This time, you are to discover the labyrinth itself...

A labyrinth is laid on a square $N \times N$ grid, and hence contains N^2 cells. Every cell holds the total number of cells that are visible both vertically and horizontally, the walls being opaque. For instance, in the picture above, the center cell (second row, second column) holds “1”, since there are walls surrounding the cell in all directions except above, where the next wall is one cell away. Similarly, the leftmost-bottommost cell (third row, first column) holds “4” since, from this cell, one sees 2 cells above, 2 other cells to the right, none below and none to the left.



Given the numbers in the cells, find the labyrinth. You may assume that the square grid is surrounded by walls and that the problem has a solution.

Input

The input file contains several test cases, each of them defined as follows:

The first line consists of number N , with $1 \leq N \leq 16$. Then come N lines, each containing the N integers of a row.

Output

For each input case, output an ASCII representation of the labyrinth. This representation is made of $2N + 1$ lines of $2N + 1$ characters. For instance the labyrinth above should be given as:

```

+---+--+
| | |
+ + + +
| | | |
+ ---+ +
| | |
+---+--+

```

Notice that walls are represented either by ‘-’ or ‘|’ depending on whether they are horizontal or vertical. Cells and lack of walls are rendered by ordinary spaces, while (potential) walls meet on ‘+’.

Write a blank line to separate the output of two consecutive cases.

Sample Input

```

2
2 2
2 2
4
5 6 3 4
3 4 2 1
5 6 5 4
1 4 2 1

```

Sample Output

```

+---+
| |
+ + +
| |
+---+

+---+--+
| | |
+ + ---+ +
| | | |
+ + + ---+
| | |
+--+ + + +
| | | |
+---+--+

```