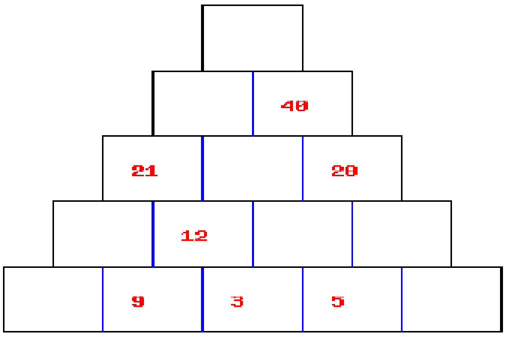
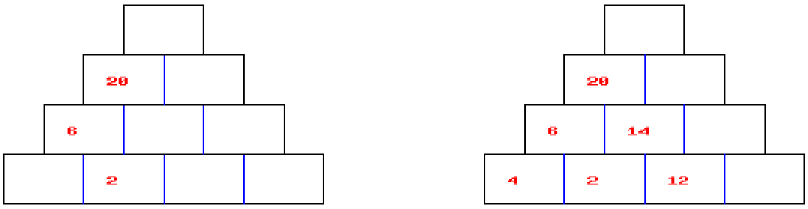
4804 eDoor

Bob the builder is a small business man with grand ideas, and the Building Education Revolution (BER) was an opportunity not to be missed. For each school hall that he has a contract to build, Bob plans to install a door that displays an educational puzzle and will only open if a correct solution is entered. The puzzle consists of a step pyramid with N rows and has a base of N boxes, as illustrated below with N equal to five (5). Each box is either empty or has a single non-negative integer whose value is:

1. larger than integers that may be present in the two boxes directly beneath it, and
2. equal to the sum of the integers, if both present, in the two boxes directly beneath it.



A puzzle is solved by making the maximum possible assignment of unique nonnegative integers, which satisfy the above mentioned rules, to empty boxes. If more than one value is possible for an empty box, then it must be left empty, as shown in the following example:



Your task is to write a program to solve such puzzles and thus verify the entered answers.

Input

Input consists of several wall puzzles. Each puzzle description begins with an integer N, on a line by itself, that represents the number of rows in the puzzle. Each of the following N lines lists the entries in one of the rows starting with the top row. The top line has one entry, and each of the following $N - 1$ lines has one more entry than the one preceding it. The entries on each line are separated by single spaces, and each entry is either the character '*', which represents an empty box, or an integer with a value in the range of 0 to 2000000000, inclusive. $0 \leq N \leq 50$, and a value of N equal to zero denotes the end of the input.

Output

For each puzzle, print the puzzle number (starting with 1, and using the format in the sample) followed by n lines showing the puzzle with integer values in all the boxes that can be resolved. Indicate boxes that cannot be uniquely resolved with an asterisk ('*').

Sample Input

```
6
*
* 100
34 * *
* * * 33
* * 10 * *
* 3 * * * 9
6
*
111 *
55 * *
* 28 * 24
* 15 13 * 9
* * 5 * * *
0
```

Sample Output

```
Puzzle 1
*
* 100
34 * *
* * * 33
* * 10 * *
* 3 * * * 9
Puzzle 2
219
111 108
55 56 52
27 28 28 24
12 15 13 15 9
2 10 5 8 7 2
```