

6451 Tables

HTML uses a simple tag format for table layout. You are to create ASCII-art tables based on a simplified notation.

Logically, a table can be considered an m by n grid, each with a 2-character wide by 1-character high cell, for example a 2×3 grid would be as follows:

```

-- -- --
|11|12|13|
-- -- --
|21|22|23|
-- -- --

```

The output consists of $2m + 1$ rows, each having $3n + 1$ characters (including leading and trailing spaces in odd-numbered rows).

But some tables are not strictly grid-based, because the values for certain cells can span multiple rows and/or columns. Here is the layout if cell 11 has a row span of 2 and cell 22 has a column span of 2:

```

-- -- --
|11|12|13|
      -- --
|  |22  |
-- -- --

```

You are to create these ASCII-art tables given m , and the row span and column span values.

Input

The input consists of multiple test cases. The first line of each test case contains one integer m , the number of rows. Then, m lines follow, describing the row and column spans in the layout. Each of the next m lines gives the span information for a row. The span information is given only for the upper left cell in the span. If N row spans and column spans need to be specified for a given row, these would be specified as:

$$N \text{ RS}_1 \text{ CS}_1 \dots \text{RS}_N \text{ CS}_N$$

where the values RS_k and CS_k are between 1 and 9, and specify how many rows and columns the next cell needing information in this row needs to occupy. Previous row spans or column spans may imply that fewer cells than the total number of columns need to be specified for a particular row.

You may assume that the input is valid:

- the number of rows and columns are between 1 and 9;
- every row will have the same number of columns;
- there are no overlapping spans of cells;
- every cell is contained in some span of cells.

There are at most 100 test cases and the end of input is indicated by $m = 0$.

Output

For each case, display the ASCII-art table, with the row-column index displayed in the top left corner of every span of cells. Display a blank line after every case.

Sample Input

```

3
2 1 1 1 1
2 1 1 1 1
2 1 1 1 1
3
2 1 1 2 1
1 1 1
2 1 1 1 1
3
1 1 2
2 1 1 1 1
2 1 1 1 1
3
2 2 1 1 2
1 2 2
1 1 1
0

```

Sample Output

```

-- --
|11|12|
-- --
|21|22|
-- --
|31|32|
-- --

-- --
|11|12|
--
|21| |
-- --
|31|32|
-- --

-- --
|11  |
-- --
|21|22|
-- --
|31|32|
-- --

-- -- --

```

```
|11|12 |  
  -- --  
| |22 |  
  --  
|31|  |  
  -- -- --
```