Google Code Jam '22 Qualification Round Problem A - Punched Cards

Time limit: 5.0s **Memory limit:** 1G

A secret team of programmers is plotting to disrupt the programming language landscape and bring punched cards back by introducing a new language called *Punched Card Python* that lets people code in Python using punched cards! Like good disrupters, they are going to launch a viral campaign to promote their new language before even having the design for a prototype. For the campaign, they want to draw punched cards of different sizes in ASCII art.

The ASCII art of a punched card they want to draw is similar to an $R \times C$ matrix without the top-left cell. That means, it has $(R \cdot C) - 1$ cells in total. Each cell is drawn in ASCII art as a period (`.`) surrounded by dashes (`-`) above and below, pipes (`|`) to the left and right, and plus signs (`+`) for each corner. Adjacent cells share the common characters in the border. Periods (`.`) are used to align the cells in the top row.

For example, the following is a punched card with $R = 3$ rows and $C = 4$ columns:

```
..+-+-+-+
..|.|.|.|
+-+-+-+-+
|.|.|.|.|
+-+-+-+-+
|.|.|.|.|
+-+-+-+-+
```

There are more examples with other sizes in the samples below. Given the integers R and C describing the size of a punched card, print the ASCII art drawing of it as described above.

Input Specification

The first line of the input gives the number of test cases, T. T lines follow, each describing a different test case with two integers R and C: the number of rows and columns of the punched card that must be drawn.

Output Specification

For each test case, output one line containing `Case #x:`, where x is the test case number (starting from 1). Then, output $(2 \cdot R) + 1$ additional lines with the ASCII art drawing of a punched card with R rows and C columns.

Limits

$1 \leq T \leq 81$.

$2 \leq R \leq 10$.

$2 \leq C \leq 10$.

Sample Input

```
3
3 4
2 2
2 3
```

Sample Output

```
Case #1:
..+-+-+-+
..|.|.|.|
+-+-+-+-+
|.|.|.|.|
+-+-+-+-+
|.|.|.|.|
+-+-+-+-+
Case #2:
..+-+
..|.|
+-+-+
|.|.|
+-+-+
Case #3:
..+-+-+
..|.|.|
+-+-+-+
|.|.|.|
+-+-+-+
```

Explanation for Sample

Sample Case #1 is the one described in the problem statement. Sample Cases #2 and #3 are additional examples. Notice that the output for each case contains exactly $R \cdot C + 3$ periods.