

# Problem 9: SOLIDWORKS is not responding

## 7+2=9 Point(s)

Problem ID: `extrusion`

Rank: 2+3

## Introduction

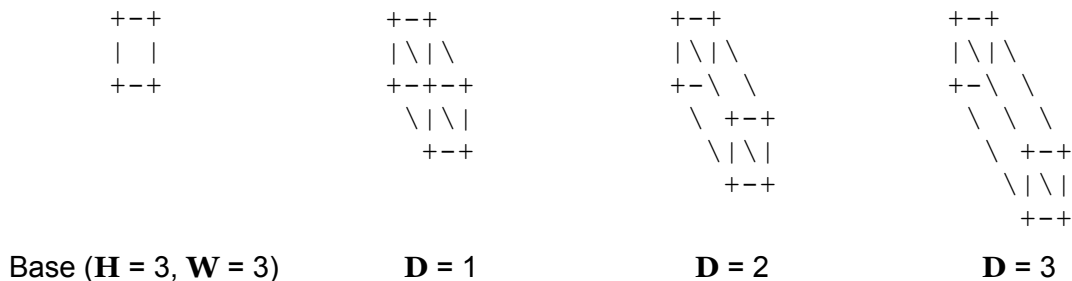
CADence is designing some 3D printed robot parts but the [CAD software](#) is [not responding!](#) (Probably because it only has 1 second runtime and 256MB RAM) Her parts are all relatively small and simple, and she has designs for them already, but she can't generate previews. Instead of wrestling with SolidWorks, she decides to write her own renderer instead!

## Problem Statement

Given a base with height **H** and width **W** as a grid of characters with **H** rows and **W** columns, output an image of the shape that is formed after being extruded to depth **D**.

The base is a closed 2D shape whose sides all meet at right angles. Every side is either horizontal or vertical. Furthermore, the base has no holes or self-intersections—only a single continuous boundary. In other words, the base is a [simple rectilinear polygon](#).

To extrude, extend the base to construct a 3D prism with **transparent faces but visible edges**. To do this, draw edges from each corner of the base towards the bottom right using backslashes `\` a total of **D** times. Each time, draw on top of the existing image. Finally, draw another base at the end to complete the prism. Here are some examples with a simple square base:



Due to the way we draw the extrusion, this results in a "perspective" such that some characters will overlap with each other when drawing. When this happens, draw only the character closer to our perspective in each position.

## Input Format

The first line of the input contains an integer **T** denoting the number of test cases that follow. For each test case:

- The first line contains three space-separated integers **H W D** denoting the height of the base, the width of the base, and the depth to be extruded, respectively.
- The next **H** lines contain **W** characters each. Together, they describe the shape of the base. Each character is one of the following:
  - A space denoting empty space
  - A plus + denoting a corner of the base
  - A pipe | denoting a vertical side of the base
  - A dash - denoting a horizontal side of the base

It is guaranteed that the base is a [simple rectilinear polygon](#) as described in the problem statement. Also, no two corners (+) are directly adjacent to one another, every corner connects exactly two sides, and the base extends to all four edges of the grid.

## Output Format

For each test case, output the image created by extruding the base to a depth of **D**. Use backslashes \ along the diagonals to show edges of the prism created by extrusion.

The judge will accept trailing spaces at the end of lines but not trailing blank lines at the end of each test case. Each line of your output should have at least one non-space character.

Note that your programming language might want you to encode the \ character as \\, as a single \ will be treated as an [escape sequence](#) instead.

## Constraints

**Time Limit: 2 seconds** (This is twice the time of a typical problem because of larger inputs!)

$$1 \leq T \leq 100$$

The total number of characters in the correct output in each test file is no more than  $10^6$  (1 MB).

### Main Test Set

$$1 \leq H, W, D \leq 150$$

### Bonus Test Set

$$1 \leq H, W, D \leq 1500$$

# Sample Test Cases

## Sample Input

[Download](#)

```

3
3 3 3
+-+
| |
+-+
7 7 2
  +-+
  | |
+-+ +-+
|   |
+-+ +-+
  | |
  +-+
7 12 1
+-----+
|         |
| +-----+ |
| |         | |
| +-----+ | |
|         | +-----+
+-----+
  
```

## Sample Output

[Download](#)

```

+-+
|\|\
+-\ \
 \ \ \
  \ +-+
   \|\|
    +-+

+-+
|\|\
+-+ \-\
|\ \ +-+
+-\ \|\|\
 \|+-+ +-+
  \|\ \ \
   +-+ +-+
    \|\|
     +-+

+-----+
|\           |\
| +-----+ |
| |\       |\ | |
| |-----+ | |
| |\|  |\-| -+ |
+-|-+-----+| \
 \|         \ |+-----+
+-----+
  
```

## Sample Explanations

For test case #1, the base is a square. The bottom right corner of the original base and the edge extruding from the top left corner overlap. Since the edge is closer to our perspective, we output that instead of the corner.

For test case #2, the base has many corners and the image has a lot of overlapping components. Be careful to ensure your perspective is correct.

For test case #3, the base is concave with a big dent and a corner touching a different side.