9. Rasterizer

Program Name: Rasterizer.java Input File: rasterizer.dat

The company you work for is in need of a simple rasterizer for a drawing program for kids. Since your boss is so impressed with your programming skills, he has asked you to implement it. A rasterizer is an algorithm that takes drawing commands and renders the 2D image they describe. Each draw command is comprised of:

- a primitive keyword indicating the shape,
- the color used, and
- the primitive specific details, such as the location, width and height for a square or the starting and ending points for a line.

Before the code goes into the application your boss wants you to write a small program to test it. You will read information from a text file and output the image as text. For simplicity the colors will just be uppercase characters A to Z. Each line will contain a single primitive keyword along with the necessary details to rasterize it as shown in the table below.

The primitive line formats you will need to support are:

Primitive Keyword and Details	Description
BOX color tlx tly brx bry fill	color: the color character tlx: integer for the x coordinate of the top left corner tly: integer for the y coordinate of the top left corner brx: integer for the x coordinate of the bottom right corner bry: integer for the y coordinate of the bottom right corner fill: `Y` if you should fill the whole box with the color, `N` if you should just draw the 1 pixel outline
LINE color x1 y1 x2 y2	color: the color character x1: integer for the x coordinate of the starting point y1: integer for the y coordinate of the starting point x2: integer for the x coordinate of the ending point y2: integer for the y coordinate of the ending point **NOTES:** Either x1 and x2 will be equal, or y1 and y2 will be equal (no diagonal lines)
CROSS color cx cy w h	color: the color character cx: integer for the x coordinate of the center of the cross cy: integer for the y coordinate of the center of the cross w: total width of the cross h: total height of the cross **NOTES:** If the width or height is even, add the extra pixel to the right or the bottom (towards the maximum)

(continued on next page)

9. Rasterizer (cont.)

Input

The first line will contain a single integer `n` that indicates the number of image descriptions to follow. For each image description:

- The first line contains 2 integers `w` and `h` indicating the width and height of the image.
- In rendering, screen resolution, position and size are done in width x height order, that is, the `x,y` coordinate of the top left of the image is always `0,0`, with `x` increasing positively to the right to `width-1` and `y` increasing downwards to `height-1`.
- Initially, the image should have all cells initialized to the period character '`.`' to denote that nothing has been drawn there.
- Read and process the commands one line at a time:
    - For the primitive keywords `BOX`, `LINE`, or `CROSS` found at the beginning of each input line, use the description from the table on the previous page corresponding to each keyword to interpret the details following the keyword on the input line. For instance, for the input line `BOX W 1 1 3 4 Y` the drawing program should draw a box from `1,1` to `3,4` filling the whole box with `W`.
    - Use the single word `END` to complete the current image by printing it.

Output

For each image you will output the image, a row per line, without any spaces between columns. There should be a blank line between images.

Example Input File

```
1
10 8
BOX W 1 1 3 4 Y
BOX A 9 7 100 100 N
LINE L 2 4 8 4
CROSS X 5 4 4 4
END
```

Example Output to Screen

```
..........
.WWW......
.WWW......
.WWW.X....
.WLLXXXXL.
.....X....
.....X....
.........A
```