5. Depth Sorted Rasterizer

Program Name: Depth.java   Input File: depth.dat

The company you work for needs a 3D rasterizer for a rendering program and you have been asked to develop 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 (i.e., shape), the color used, and the primitive specific details, such as the location, width and height for a square, for example, or the starting and ending points for a line.
- Each primitive will have a depth floating point value that indicates how deep the object is into the scene. If you think about the image as being a picture being taken from a virtual camera, the depth value 0.0 is as close as you can get to the camera, and 1.0 is as far as the camera can see.
  - Anything less than 0.0 or greater than 1.0 cannot be seen by the camera.
  - An object at a depth of 0.5 would be obscured, either completely or partially, by an object at a depth of 0.3.
  - When a part of one object overlaps a part of another object and they have the same depth value then the primitive drawn last should be the one visible.
  - Your code should handle all these cases when generating the 2D image.

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. Each line of the image description will contain a single primitive along with the necessary details to rasterize it. The primitive line formats you will need to support are described in the table below. For simplicity the colors in the image will be represented as uppercase characters A to Z.

Primitive Keyword and Details	Description
BOX color tlx tly brx bry z 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 z: the depth of the primitive fill: `Y` if you should fill the whole square with the color, `N` if you should just draw the 1 pixel outline
LINE color x1 y1 x2 y2 z	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 z: the depth of the primitive **NOTES**: Either x1 and x2 will be equal, or y1 and y2 will be equal (no diagonal lines)
DIAMOND color cx cy s z	color: the color character cx: integer for the x coordinate of the center of the diamond cy: integer for the y coordinate of the center of the diamond s: the number of pixels the diamond extends in each direction from the center (so if it is 3, the diamond goes 3 to the left, 3 to the right, 3 up, and 3 down, making the distance of the longest points 7) z: the depth of the primitive **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)

5. Depth Sorted Rasterizer (cont.)

Input

The first line will contain a single integer n that indicates the number of image descriptions to follow. Each image description will be composed of an unknown number of lines as follows:

- 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 0.1 Y the drawing program should draw a box from 1,1 to 3,4 filling the whole box with W at a depth of 0.1.
  - 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. Pixels in which nothing was drawn should be indicated by the period character (.).

Example Input File

```
1
10 8
BOX W 1 1 3 4 0.1 Y
BOX A 9 7 100 100 0.5 N
LINE L 2 4 8 4 0.6
DIAMOND X 5 4 3 0.7 N
END
```

Example Output to Screen

```
..........
.WWW.X....
.WWWX.X...
.WWW...X..
.WWWLLLLL.
...X...X..
....X.X...
.....X...A
```