In a nutshell, this assignment extends the previous one to allow an arbitrary number of rectangles to be drawn onto the canvas (instead of just one).

In more detail, instead of the input consisting of four nonnegative integers as in the previous assignment, the input now consists of $4r$ nonnegative integers, for arbitrary $r \geq 0$. As before, there is an arbitrary amount of nonempty whitespace (including spaces, tabs, newlines, etc.) separating these numbers. Conceptually, these $4r$ numbers describe r rectangles, indexed 0 through $r - 1$, with rectangle k described by the four numbers starting with the $4k$-th (0-indexed) one of the input. Each rectangle is specified using the same method as the previous assignment. That is, the four numbers corresponding to a rectangle determine, in order, the 0-indexed column and row of its top-left corner, its width (number of columns) and its height (number of rows).

If (and only if) the program is invoked with a command-line argument `D` then it should present output in a **debug mode**. This mode results in a period (`.`) being printed where the non-debug mode prints a space. This mode also results in printing of the coordinate axes. The horizontal position (x coordinate) is marked using two lines at the very top and the vertical position (y coordinate) is marked using two columns at the very left. Further details of these axes may be inferred by examining the sample outputs below (which are presented in debug mode).

All the **other details** of this assignment, including source code quality, comments, documentation (README), sample inputs and outputs, packaging, submission, proper use of standard streams, allowable use of resources, attribution, etc., are the *same as those of the previous* one. However, it will be graded more strictly than was the previous one on those aspects. For instance, poorly formatted or poorly documented code will result in more substantial loss of points. These details are therefore omitted here for brevity. As always, the class discussion forum and in-class discussions should be used for clarifications.

Following are some illustrative sample input-output pairs.

1. Empty input should produce an empty canvas as output.

   Input: *[empty]*
   Output: [In non-debug mode, exactly 24 newline characters and nothing else. In debug mode, just the coordinate axes in the format depicted later.]

2. Zero-width and/or zero-height rectangles should have no effect on canvas.

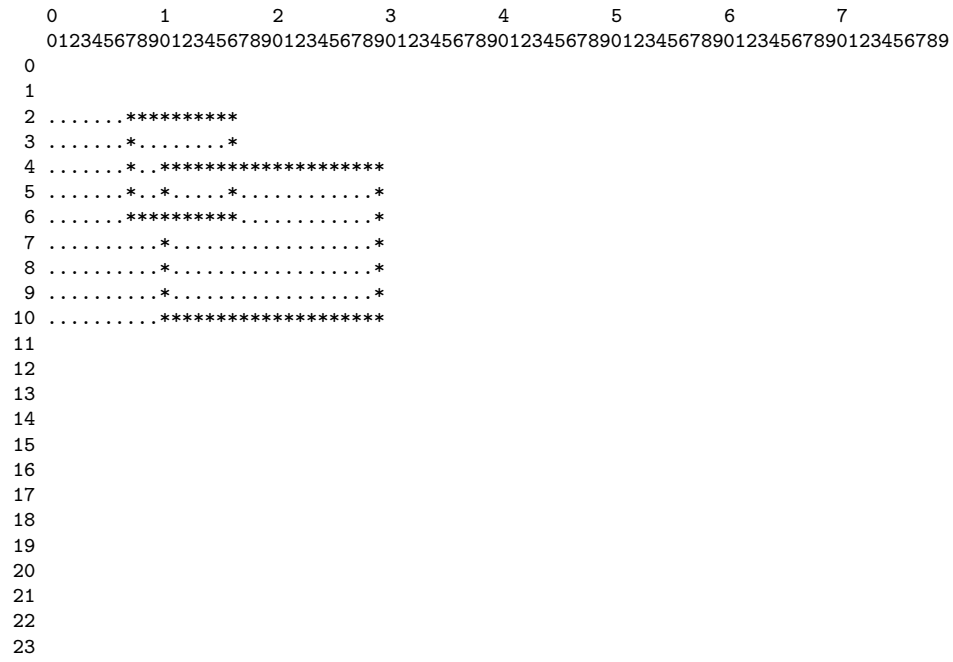
   Input: 4 2 0 11 1 7 7 0
   Output: [Same as previous output.]

3. Every valid input for the previous assignment is also valid here and should produce the same output as the previous program.

   Input: `33 6 33 9` [Same as the sample input from the previous assignment]
   Output: [Corresponding sample output from the previous assignment; omitted here for brevity. However, the debug mode should be honored here.]

4. Two rectangles with nonempty intersection.

   Input: 10 4 20 7 7 2 10 5
   Output:



5. Several rectangles.

   Input: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 3
   3 1 4 1 5 9 4 2 6 0 2 3 78 17 2 4 0 0 70 22
   Output: