Introductory PL/0 Exercise

The PL/0 environment does not save, so a text editor window is useful.

`http://ranger.uta.edu/~weems/NOTES3302/BASELINE/` has `baseline.zip` to get all files locally.

`http://ranger.uta.edu/~weems/NOTES3302/BASELINE/DIAGRAMS/` includes:

> `states.jpg` which describes the "user experience" as a state machine.
>
> Other .jpg files for the PL/0 syntax. These were produced by `http://www.graphviz.org`, a free layout package.

`http://ranger.uta.edu/~weems/NOTES3302/BASELINE/EXAMPLES/` includes PL/0 code to be pasted into the "PL/0 Code:" textarea of `http://ranger.uta.edu/~weems/NOTES3302/BASELINE/pl0.baseline.html`

All boxes are resizable, except the canvas.

1. Start by pasting, compiling, and running the first example (7 lines) of `test.txt`. You should get an exception. Fix the situation by entering three values in "Integer Input:" - an x and y for the circle's center and a radius. The "Mouse Position (cvx, cvy);" is useful for understanding the canvas's coordinate system.

Put a 1 in "Step Limit", hit Run and use Continue to single step. Observe how the "Run-Time Stack:" changes. Also the value of the program counter, `p`, can be used to find the relevant code line in "Listing:".

2. Paste, compile, and run (without stepping) the second example from `test.txt`.
Now insert after "i:=i+1" the statements "out:=i" and "call stop". Compile and run.

3. Paste, compile, and run `binsearch.txt`. Observe that `a` is an array of integers and a procedure may have integer values (but not arrays) passed to it.

Add a `call` to `stop` so you can see the contents of array `a` after it is initialized.

4. Paste, compile, and run `crosshair.txt`, which uses the canvas. `cvx` and `cvy` are the coordinates of the cursor, but will be -999999 when the cursor is outside the canvas. Use Stop and Continue.

5. Write a program to draw a number of polygons (no more than 10) on the canvas using the built-in procedure `cvline`. You should read the input from "Integer Input:". The first line is the number of polygons. Each polygon will be given as a number of points (at least 3, no more than 10) followed by the x and y coordinates.

6. Modify your program from 5. to have a ball that moves from point to point. You will need to use arrays and should redraw everything for each movement of the ball.