

## Bresenham lines and ellipses. Deadline: Nov 2 midnight.

Write a program which draws lines and axis-aligned ellipses specified by the input file using the Bresenham's algorithm. Thus, if no antialiasing is used, your programs will operate only on integers. To draw antialiased primitives, you can only use floats to compute the intensity for the each of the two pixels on the scanline you will be shading. Draw white lines on a black background.

Format of the input file (to be read from the standard input – use `scanf()`):

```
< n >
< primitive_code1 >< data1 >
< primitive_code2 >< data2 >
.....
< primitive_coden >< datan >
```

We will be using four primitive codes:

- l**: line with no antialiasing
- L**: antialiased line
- e**: ellipse with no antialiasing
- E**: antialiased ellipse

An 'l' or 'L' code will be followed by four integers, which your programs will interpret as endpoints of the line to be drawn. An 'e' or 'E' code will be also be followed by four integers, the coordinates of the center of the ellipse and the length of its horizontal and vertical axes. If the numbers which follow are  $c_x$ ,  $c_y$ ,  $a$  and  $b$  then you will need to draw the ellipse centered at  $(c_x, c_y)$  and with axes  $2a$  and  $2b$ , like the one shown in Figure 1. The equation of that ellipse, which you will need to design your decision variable, is

$$\frac{(x - c_x)^2}{a^2} + \frac{(y - c_y)^2}{b^2} = 1.$$

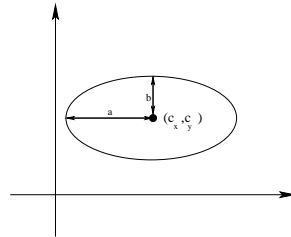


Figure 1: An ellipse specified by the data  $c_x$ ,  $c_y$ ,  $a$  and  $b$ .

Grading: 25% per primitive. No credit for algorithms using floats to draw unantialiased lines and ellipses. 10% total for clarity of the code. 20% for a reasonably optimized code (no expensive/unnecessary operations etc).