Program AreaCalc

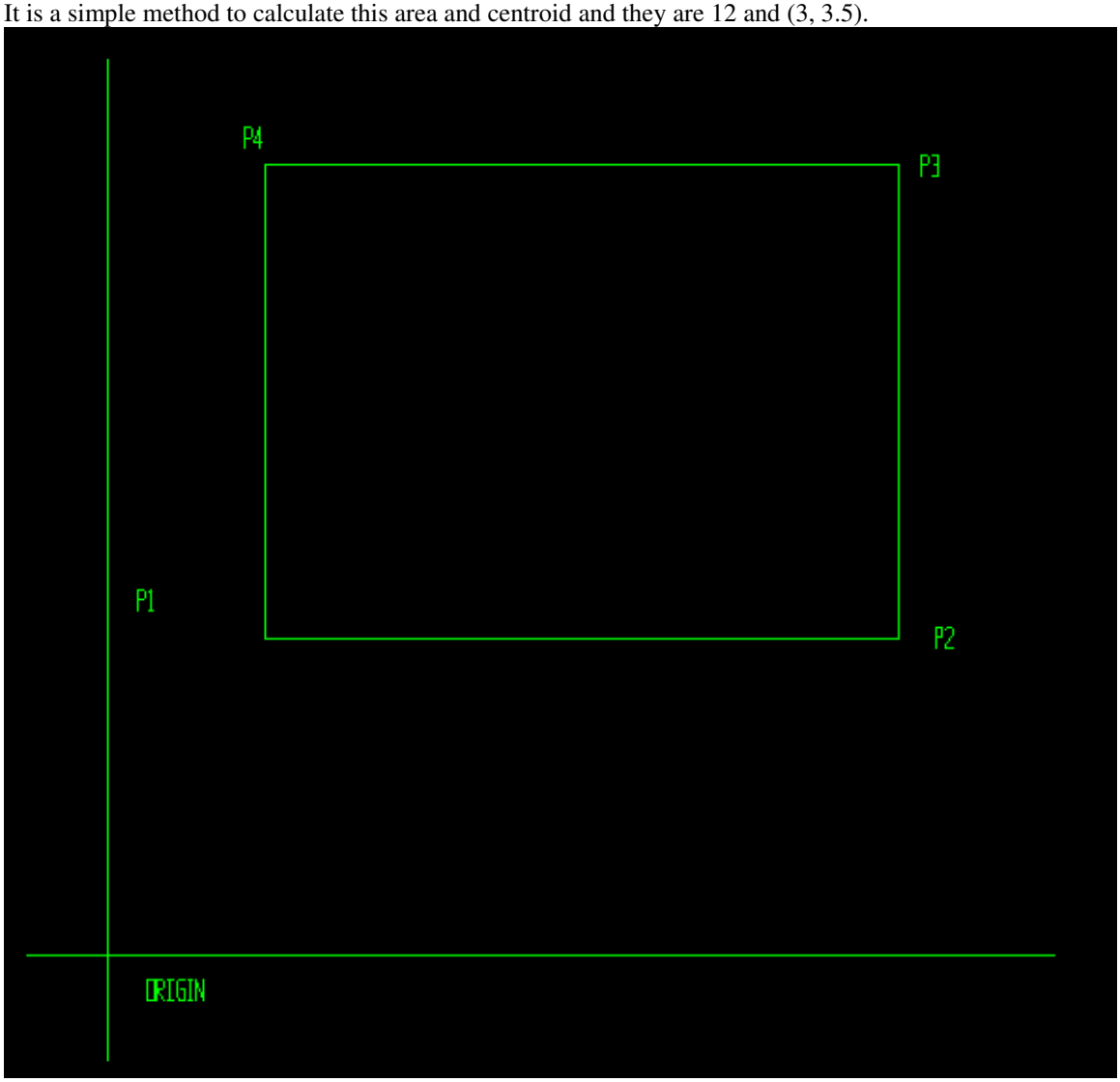
This program returns the bounded area and coordinates of the centroid of any shape. You type the coordinates of the tiepoints into a matrix and execute the program. It then uses a neat trick with cross and dot products to calculate the area and centroids. Essentially, it steps through the tiepoints you type in and makes vectors from the origin to the tie points. As it goes in order around the shape, if the first vector to the second is a clockwise sweep, it subtracts the area between the vectors. If it is a counter-clockwise sweep, it adds the area. When all is said and done, you are left with the shape you have defined. Since it is adding and subtracting triangles, there is a limitation to shapes that are not made from straight edges. If all the edges are straight, it will yield exact results. If there is a curve, then the more points you are willing to type on the curve, the more accurate your answer.

Execution

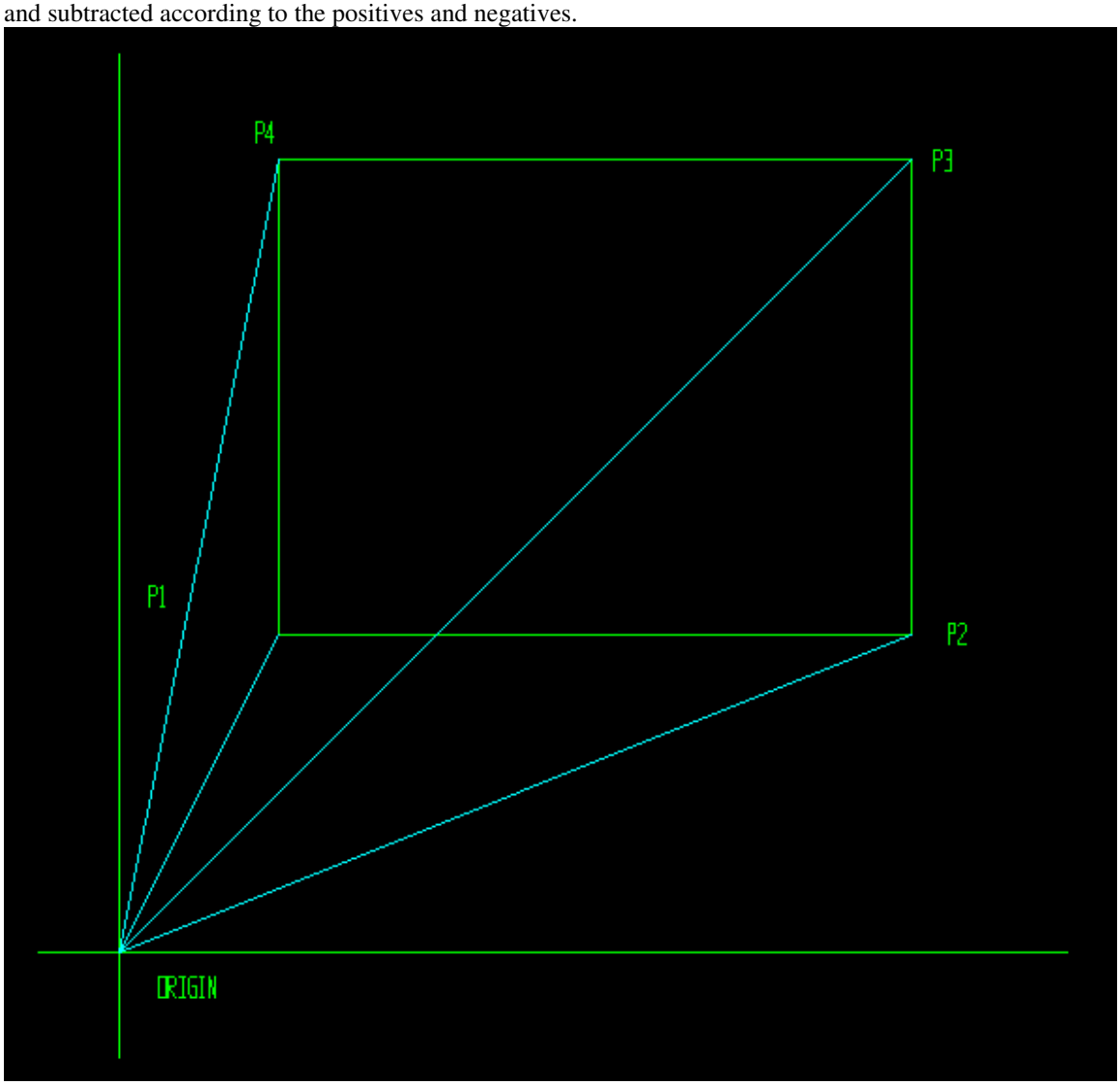
Just execute the program. It will create a matrix for you and tell you to edit it. You open the matrix with the matrix editor and type in all the x and y coordinates. Re-run the program, and when it sees the completed matrix, it will calculate the area and centroid and draw the shape on the screen.

Example 1: A simple rectangle.
If you want to know the area and centroid of this rectangle, all you do is type in the coordinates of the points. (Centroid will be given with respect to the chosen origin). For this example, the coordinates are P1(1,2), P2(5,2), P3(5,5), P4(1,5). You do not have to type in P1 again since the program assumes connectivity from the last tie point to the first.

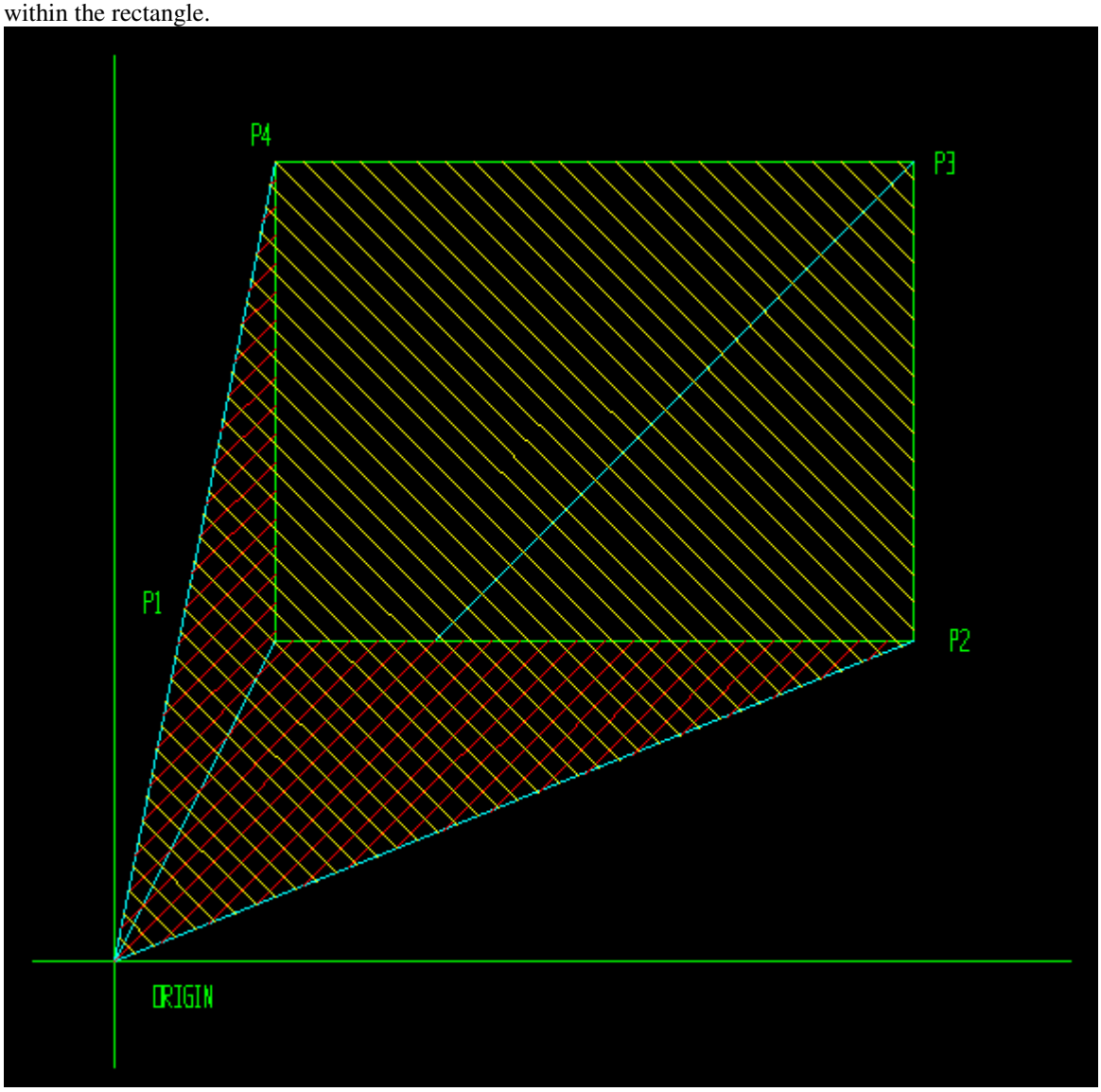
It is a simple method to calculate this area and centroid and they are 12 and (3, 3.5).



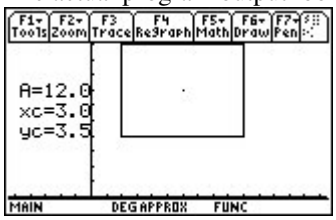
The program then steps through the vectors from the origin: OP1, OP2, OP3, OP4, OP1 again. P1 to P2 is a clockwise rotation, so it is a “negative”. P2 to P3 and P3 to P4 are CCW so they are “positives”. And P4 to P1 is a negative again. The areas bounded by the shape edge and the two vectors (triangles) are added and subtracted according to the positives and negatives.



A representation of the triangles is shown below. The yellow, 135 deg lines represent the positives and the red 45 deg lines represent the negatives. For the areas outside the rectangle between the origin, you can see that whatever is added by the positives is then removed by the negatives. Leaving only the shaded area within the rectangle.



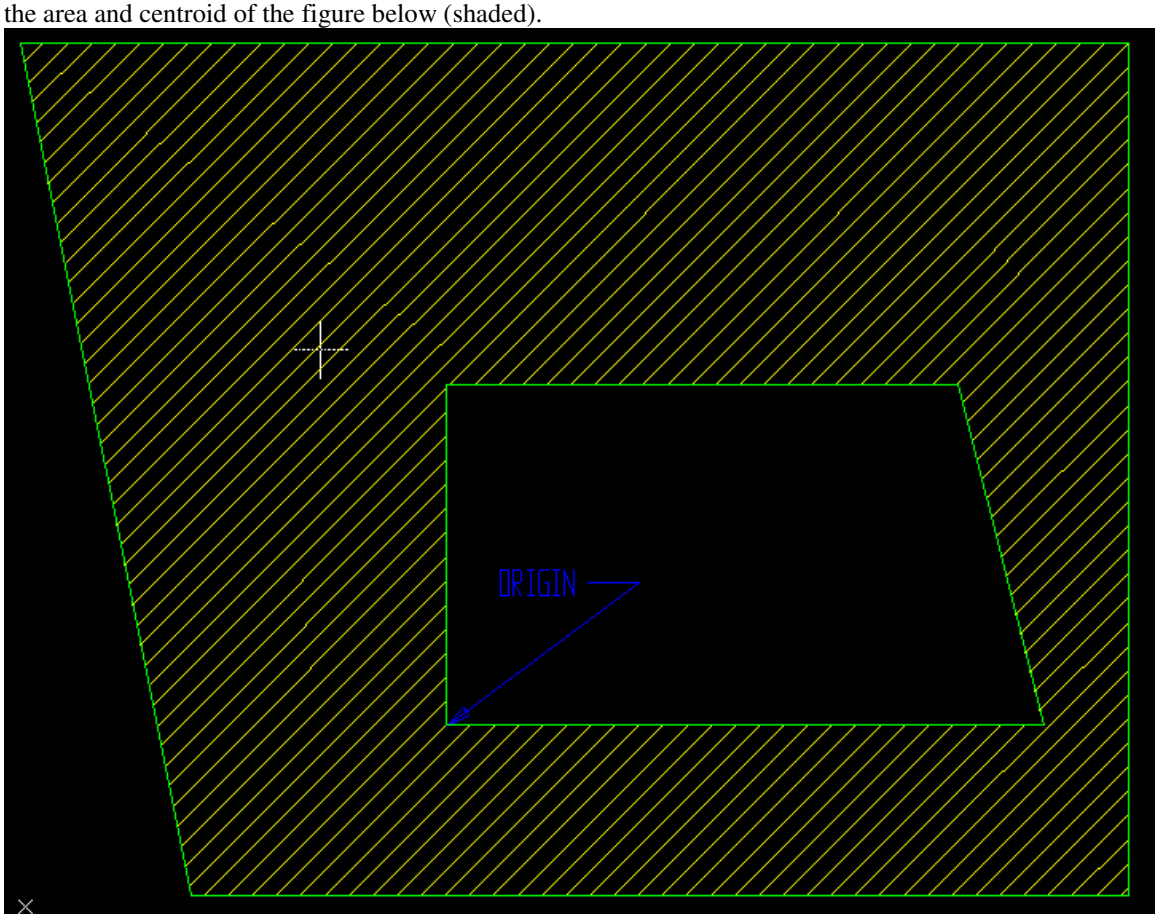
The actual program output looks like this:



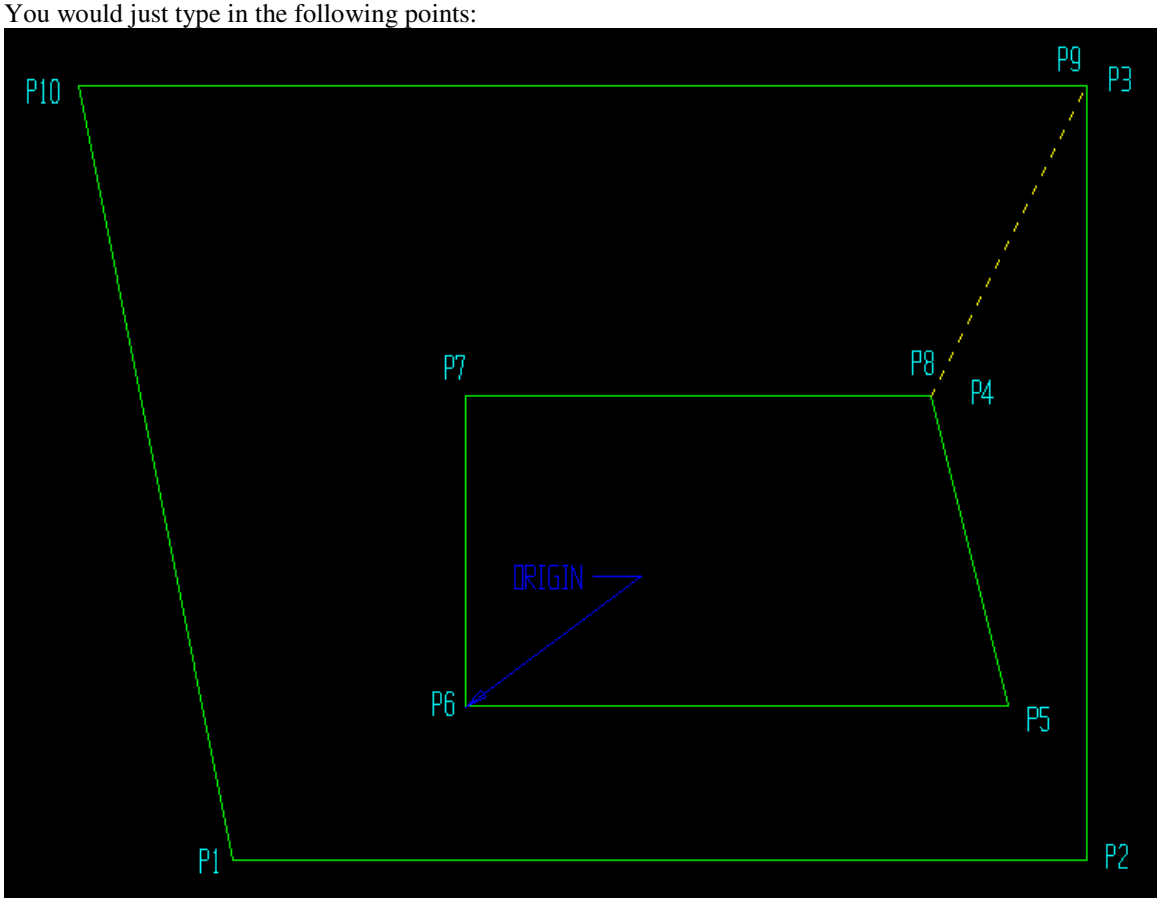
The program draws the shape (for reference), places a small dot on the centroid, and reports the area and centroid coordinates. (These are also stored in variables after execution).

Example 2: A shape with a hollow

The program can handle hollow shapes, as long as a connectivity path is established. Suppose you wanted the area and centroid of the figure below (shaded).



You would just type in the following points:



Notes:

- A line has been added from the upper right corner of the object to the upper right corner of the hole. This is so that all the edges have a continuous path (required for proper operation).
- Pay close attention to the order of the points. You must go …2, 3, 4, 5…. You CANNOT go ….2, 3, 8, 7….. If you do, you will "cross the streams" as it were and the program will give you bad results. To keep it straight, you must imagine the dotted line as a gap (which it actually is as far as the program is concerned, but its width = 0, so no gap). If it were a gap, you could not go from P3 to P8 to P7 since you would have to cross open space. It does not matter that P4 & P8 and P3 & P9 are coincident points. The order of the tiepoints is critical to the function of the program.
- Walk through the CW (negative) and CCW (positive) triangles in your mind and you will see that the area left is the area you want.

If you want to try this example, the points are (remember, you don't have to add P1 again at the end, but it will not crash if you do): Sample output from the program is to the right.

P	X	Y
1	-15	-10
2	40	-10
3	40	40
4	30	20
5	35	0
6	0	0
7	0	20
8	30	20
9	40	40
10	-25	40