



Defn:

Zone Division

The line drawing code for each individual zone is not optimized, since, we need to write a lot of redundant lines of code. Hence, we use 8-way symmetry.

Explanation

If a ^{point} ~~line~~ is selected at zone 0, as (x, y) then we can derive its corresponding coordinates in the other zones by using reflection in the lines $(y=x)$ & $(y=-x)$.

From the above figure it can be determined that any point in another zone can be traced back to zone 0 (x, y) pattern, by going through the corresponding rearrangement of the x & y coordinates in the other zones.

How algo works.

So, if we can derive the mid point line drawing algorithm for zone 0. We can use it draw a line in any other zone, just by rearrangement of the coordinates in any other zone as shown in figure above.

So we transform the points to zone 0. find the next pixel and by rearrangement, draw the pixel in its actual zone.

Converting from any other Zone to Zone 0:

	<u>For x</u>	represents the current value of the coordinate in their respective zone.	<u>For y</u>
Zone 0	New $x = x$		New $y = y$
Zone 1	New $x = y$		New $y = x$
Zone 2	New $x = y$		New $y = -x$
Zone 3	New $x = -x$		New $y = y$
Zone 4	New $x = -x$		New $y = -y$
Zone 5	New $x = -y$		New $y = -x$
Zone 6	New $x = -y$		New $y = x$
Zone 7	New $x = x$		New $y = -y$

if there is a minus sign, it needs to be multiplied to the current coordinate to get the new coordinate

↑
The conversions are almost similar except for Zone 2 & Zone 6
↓

Converting from Zone 0 to any other Zone:

	<u>For x</u>	<u>For y</u>
Zone 0	New $x = x$	New $y = y$
Zone 1	New $x = y$	New $y = x$
Zone 2	New $x = -y$	New $y = x$
Zone 3	New $x = -x$	New $y = y$
Zone 4	New $x = -x$	New $y = -y$
Zone 5	New $x = -y$	New $y = -x$
Zone 6	New $x = y$	New $y = -x$
Zone 7	New $x = x$	New $y = -y$