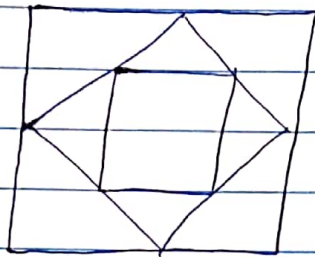


Assignment → 1

- ⇒ Problem statement : write a C++ program to draw the following pattern using line drawing. also use Bresenham's for square and DDA for diamond



- ⇒ Objective - we should be able to draw and study different line drawing algorithms
- ⇒ Outcome - To be able to draw different shapes using Bresenham's and DDA line drawing algorithms
- ⇒ Software reqt - 64 bit OS (fedire)!, at least
- ⇒ Theory → pixel - it is the smallest unit of memory which are to be displayed on a screen
- Raster-form - method of displaying an image in terms of 2 arrays of pixels
 - Frame buffer - It is a complete memory buffer which contains data of each pixel to be displayed next
- DDA - Different ~~A~~ Digital analysis
- In this method, we set initial value for pixels to be displayed on the screen
 - These pixels are incremented every iteration according to the slope

The slope can be calculated with the end point of the given line as

$$M = \frac{y_2 - y_1}{x_2 - x_1}$$

Bresenham's Algorithm - This method does not involve slope calculation rather used calculation of 'error factor' which may be added to the pixels position every iteration. This method does not have floating point calculation.

- ⇒ Advantages → DDA is a simple code to calculate pixel position
- It is easy to understand
 - Can be implemented with basic math knowledge
 - Bresenham's algorithm runs faster than DDA
 - Uses only addition and subtraction
 - No need of special function to type cast to integers

- ⇒ Disadvantages → DDA involve floating point operation
- Due to round off errors are introduced
 - Bresenham's algorithm is really complex algorithm to implement.

⇒ Test cases →

| | Input | Expected o/p | Actual o/p |
|----|--|--|------------|
| 1) | corner point of Square (0,0) (100,100) | Inner square side = 50 outer square side = 100 | Success |
| 2) | corner point of Square (0,0) (200,200) | Inner square side = 100 outer square side = 200 | Success |

⇒ conclusion → Thus we are able to generate a given figure using Bresenham's and PDA algorithm.