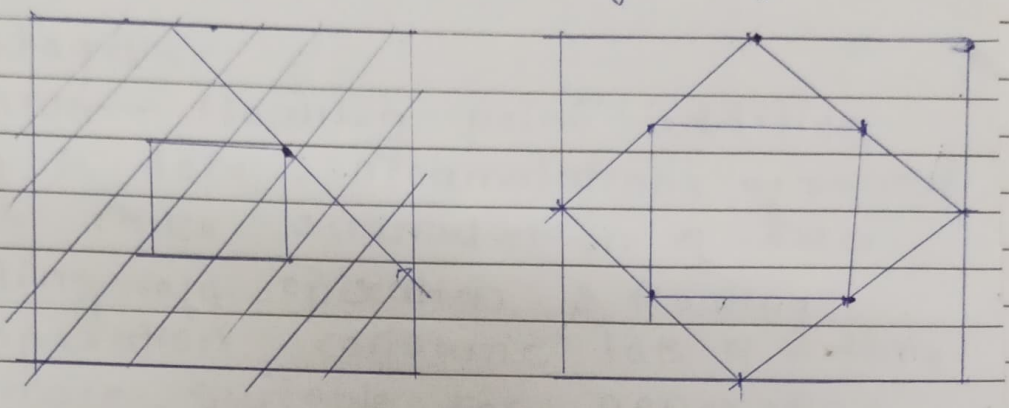
Page No.
Date

* Assignment No. 2 *

Aim - To write a C++ program to draw the following pattern, using DDA and Bresenham's line drawing algorithm.



Theory :-

① DDA line Drawing Algorithms :-

- DDA refers to Digital Differential Analyzer
- DDA line drawing algorithm is used to draw line on a screen in an incrementally. The algorithms is called the Digital Differential Analyzer because it interpolate point based on the difference between the start and end points.

Advantages :-

1) It is a faster method than method of using direct use of line equation.
2) This method does not use multiplication theorem.
3) It allows us to detect the change in the value of x and y, so plotting of same point twice is not possible.

Page No.
Date

4) This method does not gives overflow indicate when a point is repositioned.
5) It is an easy method because each of involve just two additions.

Disadvantages:-
1) It involves floating point addition rounding is done. Accumulations of round off error cause accumulation of error.
2) Rounding off operation & floating point operation consume lots of time
3) It is more suitable for generating line using the software. But it is less suited for hardware implementation.

② Bresenham's line Drawing Algorithm.
- This algorithm is used for Scan Converting a line. It was developed by Bresenham.
- It is an efficient method because, it involve only integer addition, substraction & multiplication operation. These operation can be perform very rapidly so lines can be generated quickly.

Advantages:-
1) It involves only integer arithematic, so it is.
2) It avoids the generation of duplicate points.

Page No.
Date

3) It can be implemented using hardware because it does not use multiplication & division.

4) It is faster as compared to DDA because it does not involves floating point calculations like DDA Algorithm.

Disadvantage :-

1) This algorithm is meant for basic line drawing only initializing is not a part of Bresenham's line algorithm. So to draw smooth lines, you should want to look into a different algorithm.

DDA Algorithm	Bresenham's Algorithm.
① It uses floating point.	① It uses fixed point.
② less efficient.	② More efficient.
③ less calculation speed	③ More calculation speed
④ It is costlier.	④ It is cheaper.
⑤ less precision or accuracy.	⑤ More precision or accuracy.
⑥ Complex calculation	⑥ Simple calculation
⑦ optimization is not provided	⑦ optimization is provided.

(7)
(4)

Page No.
Date

Algorithm:-

① DDA line Drawing Algorithm:-

step 1- Start Algorithm.

step 2- Declare $x_1, y_1, y_2, x_2, dx, dy, x, y$ integer variables.

step 3 - Enter value of x_1, y_1, x_2, y_2.

step 4 - Calculate $dx = x_2 - x_1$

step 5 - Calculate $dy = y_2 - y_1$

step 6 - If ABS (dx) > ABS (dy)
Then step = abs (dx)
Else,
step = abs (dy)

step 7 - $x_{inc} = dx / step$
$y_{inc} = dy / step$
assign $x = x_1$
assign $y = y_1$

step 8 - Set pixel (x, y)

step 9 - $x = x + x_{inc}$
$y = y + y_{inc}$
Set pixel (Round (x), Round (y))

Page No.
Date

step 10 - Repeat step 9 until $x = x_2$

step 11 - End Algorithm -

⑥ Bresenham's line Drawing Algorithm.

step 1 :- start Algorithm

step 2 - Decalare variable
$x_1, x_2, y_1, y_2, d, i_1, i_2, dx, dy$

step 3 - Enter the value of x_1, y_1, x_2, y_2
(x_1, y_1) = starting point
(x_2, y_2) = ending point.

step 4 - Calculate $dx = x_2 - x_1$
$dy = y_2 - y_1$
$i_1 = 2dy$
$i_2 = 2(dy - dx)$
$d = i_1 - dx$.

step 5 - Consider (x, y) as starting point and x_{end} as maximum possible value of x.
If $dx < 0$
Then $x = x_2$
$y = y_2$
$x_{end} = x_1$

(7)
(6)

Page No.
Date

If $dx > 0$
Then $x = x_1$
$y = y_1$
$x_{end} = x_2$

step 6 - Generate point at (x, y) co-ordinates.

step 7 - Check if whole line generated
If $x >= x_{end}$
Stop.

step 8 - Calculate co-ordinates of the next pixel.
If $d < 0$
Then $d = d + i_1$,
If $d >= 0$
Then $d = d + i_2$
Increment $y = y + 1$

step 9 - Increment $x = x + 1$

Step 10 - Draw a point of latest (x, y) coordinates.

step 11 - Go to step 7

step 12 - End Algorithm.