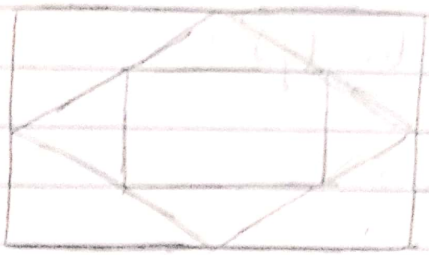
Assignment - 1 - A1

Problem Statement: Write a C++ program to draw the following pattern using line drawing algorithms. Use Bresenham's line drawing algorithms for square and DDA line drawing drawing algorithm.



Objective : To draw the above pattern using DDA line and Bresenham line drawing algorithm

Outcome : 1. Able to implement different line generation algorithms.
2. Able to understand different equations of line.

S/W & Hardware : Qt Creator, Fedora OS

Algorithms :

1) DDA line
DDA → Digital Differential Analyser
This algorithm is based on incremental methods.

Steps:-

1. Take input (x_1, y_1) & (x_2, y_2)

2. $dx = (x_2 - x_1)$, $dy = (y_2 - y_1)$

3. if (abs(dx) ≥ abs(dy))
   len = abs(dx)
   else
   len = abs(dy)

4. xinc = dx/len
   yinc = dy/len

5. x = x1 , y = y1
   i = 0
   while (i < len)
   {
   setpixel (x, y, color)
   x = x + xinc.
   y = y + yinc
   i++
   }

Advantages of DDA:-

1. It is the simplest line algorithm.
2. It avoids multiplication with slope.
3. It is a faster method for calculating pixel positions than the direct use of equation $y = mx + b$.

Disadvantages.

1) Last pixel accuracy is less.
2) Floating-point arithmetic in DDA algorithm is still time consuming.
3) Because of round off, errors are introduced and cause the calculated pixel position to drift away from the true line path.

- Bresenham's Line Algorithm :- is a line algorithm that determine the points of an n-directional raster that should be selected in order to form a close approximation to a straight line between two points.

Algorithm :-

1) Read (x_1, y_1) & (x_2, y_2)

2) $dx = abs(x_1 - x_2)$
   $dy = abs(y_1 - y_2)$

3) $x = x_1$, $y = y_1$, $i = 0$

4) if $(dx \geq dy)$
   $p = 2dy - dx$
   while $(i \leq dx)$
   setpixel $(x, y, color)$
   if $p < 0$
   $p = p + 2dy$
   else

```
{ p = p + 2dy - 2dx
  y = y + 1 * sign(y2 - y1)
 }
x = x + 1 * sign(x2 - x1)
 i++
end while
 end if
else
 {
   P = 2dx - dy
   while (i <= dx)
    set pixel (x, y, color)
    if  P < 0
     P = p + 2dx
    else
    {
    P = p + 2(dx - dy)
     x = x + 1 * sign(x2 - x1)
    }
    y = y + 1 * sign(y2 - y1)
      i++
   end while.
 }
```

```
int sign (int q)
{
  if (q < 0)
return -1
 else
return 1
}
```

Advantages:

- It is easy to implement
- It is fast and incremental
- The points generated by this algorithm are more accurate than DDA Algorithm.

Disadvantages:

- This algorithm is for the basic line drawing.
- Though it improves the accuracy of generated points but still the resulted line is not smooth.

Examples:

1) DDA

$x_1 = 0 \quad y_1 = 0 \qquad x_2 = 5 \quad y_2 = 3$

$dx = 5 \qquad dy = 3$

Since $dx \geq dy$

len = 5

$x_{inc} = \frac{5}{5} = 1 \qquad y_{inc} = \frac{3}{5} = 0.6$

x	y	Plot
0	0	(0, 0)
1	0.6	(1, 0)
2	1.2	(2, 1)
3	1.8	(3, 1)
4.	2.4	(4, 2)
5	~~3~~	(5, 3)