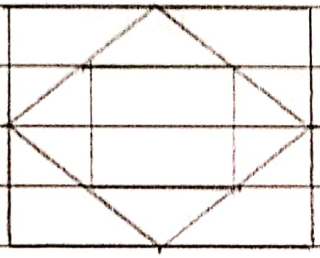
Assignment No. 7 (A2) DATE:

Problem Statement:

Write a C++ program to draw the following pattern using line drawing algorithm. Use Bresenham's line drawing algorithm for diamond.



Learning objective:

To understand and implement DDA line algorithm and bresenham's algorithm.

Learning outcome:

Students will be able to implement the concept of computer graphics and different types of line drawing algorithm, will also be able to use QtCreator

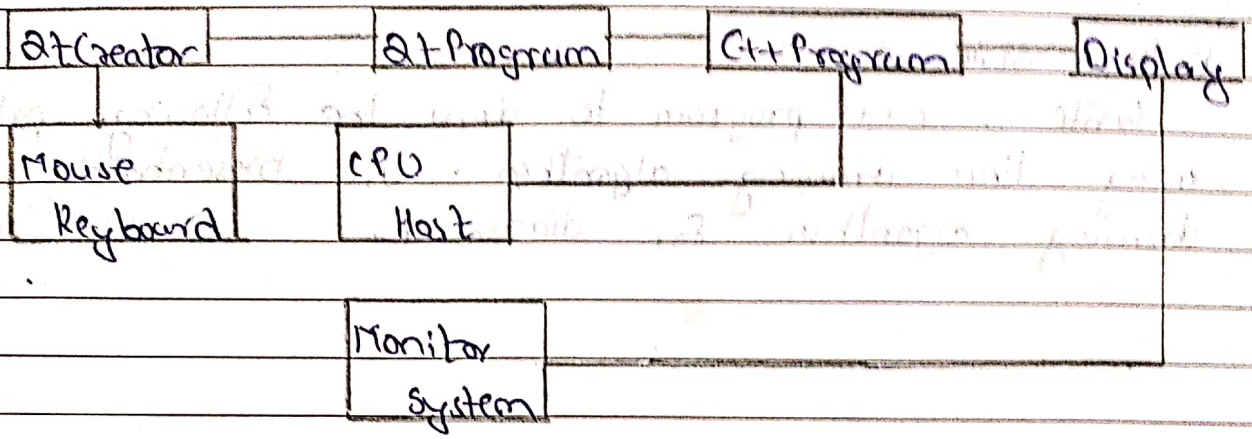
Requirements:

- Fedora 20
- QtCreator

Concepts related theory:

QtCreator is a platform (C++, java, javascript and QML) integrated development environment which is part of the SDK for the Qt GUI application development framework

DATE:



* DDA

Pseudo-code

```
DDA (int x, int y, int x2, int y2)
{ dx = x2-x1;
  dy = y2-y1;
    if (abs(dx)>=abs(dy))
        length = dx;
    else
        length = dy;
  xinc = dx/length;
  yinc = dy/length;

  set x = x1 + 0.5*sign(dx);
  set y = y1 + 0.5*sign(dy);

  for( i=1 to length)
  { setpixel (x,y);
    x = x+xinc;
    y = y+yinc;
    setpixel(x,y);
  }
}
```

DATE:

* Advantages:
- Faster for calculating pixel position.
- Easy to understand

* Disadvantages:
- Floating point operations are line consuming.

* Bresenham's Line Drawing Algorithm:
- It determines the points at an n-dimensional that should be selected in order to form an approximation to a straight line between two points.

Pseudo-Code:

```
Bresenham's (int x1, int y1, int x2, int y2)
{ x = x1; y = y1;
    dx = abs(x2 - x1)
  s1 = sign(y2 - y1); s2 = sign(x2 - x1);
    if (dx < dy)
    { temp = dx;
      dx = dy;
      dy = temp;
    }
    else
    { interchange = 0;
      e = dx - dy . dy;
      for (i = 1 to dx)
      do
        { set pixel (x, y)
          while (e >= 0)
```

DATE:

```
} if(interchange =1)
    x = x + s1;
  else
    y = y + s2
    e = e - 2*dx;
  }
if (interchange == 1)
    y = y + s2;
  else
    x = x + s1
    e = e + 2*dy;
}
```

* Applications:
- Line drawing algorithms are useful for a efficient in continuous drawing with same intensity.
- Computer Aided Design for engineering and architecture system, etc.
- Animations are useful for testing performance.

* Test Cases:

	Input	Expected o/p	Actual o/p	Result
1)	DDA $x_1 = y_1 = 0$ $x_2 = y_2 = 10$	vertical line drawn	Vertical line drawn	Pass
2)	DDA $x_1 = 9, y_1 = 10$ $x_2 = y_2 = 10$	Horizontal line drawn	Horizontal line drawn	Pass
3)	Bresenham $x_1 = y_1 = 0$ $x_2 = y_2 = 10$	Diagonal line drawn	Diagonal line drawn	Pass

DATE:

* Working:

L = 200, B = 100, x = 30, y = 40

Bresenham (50, 50, 250, 50)

Bresenham (30, 50, 50, 150)

Bresenham (50, 150, 250, 150)

Bresenham (250, 50, 250, 350)

DDA (50, 100, 150, 100)

DDA (150, 50, 250, 100)

DDA (250, 100, 150, 150)

DDA (150, 150, 50, 100)

Bresenham (100, 75, 200, 75)

Bresenham (100, 75, 200, 75)

Bresenham (100, 125, 200, 125)

Bresenham (100, 125, 100, 75)

Conclusion:

We have successfully implemented DDA and Bresenham Drawing algorithm