# ASSIGNMENT 2

## Bresenham's Algorithm

Source Code:

```
#include <iostream>
#include <GL/gl.h>
#include <GL/freeglut.h>
#include <bits/stdc++.h>
using namespace std;
void displayPoint(int x, int y){
   glColor3f(0,1,0);
   glBegin(GL_POINTS);
   glVertex2i(x,y);
   glEnd();
}

void BSAlgoLine(float x1,float x2,float y1,float y2){
   int dx=x2-x1;
   int dy=y2-y1;
   int xchange=1;
   int ychange=1;
   int x=x1;
   int y=y1;
   int p= (2*dy)-dx;
   if(x1>x2) xchange=-1;
   if(y1>y2) ychange=-1;
   if(dx==0){
       while(y<=y2){
           displayPoint(x,y);
           y=y+ychange;
       }
   }
   else if(dy==0){
       while(x<=x2){
           displayPoint(x,y);
           x=x+xchange;
       }
   }
   else if(dx>dy){
       while (x<=x2)
```

```
        {
            displayPoint(x,y);
            x=x+xchange;
            if(p<0) p=p+2*dy;
            else if(p>0){
                p=p+(2*(dy-dx));
                y=y+ychange;
            }
        }
    }
    else if(dx<dy){
        while(y<=y2){
            displayPoint(x,y);
            y=y+ychange;
            if(p<0) p=p+2*dx;
            else if(p>0){
                p=p+(2*(dx-dy));
                x=x+xchange;
            }
        }
    }

    glFlush();

}

void simpleLine(float x1,float x2,float y1,float y2){
    float step;
    float dx=x2-x1;
    float dy=y2-y1;
    step= max(abs(dx),abs(dy));
    float xin=dx/float(step);
    float yin=dy/float(step);
    float x=x1;
    float y=y1;
    for(int i=0;i<=step;i++){
        displayPoint(x,y);
        x=x+xin;
        y=y+yin;
    }
    glFlush();

}
```

```
void primitive(void){
   glClearColor(0.0,0.0,0.0,0.0);
   glClear(GL_COLOR_BUFFER_BIT);
   gluOrtho2D(0,600,0,600);
   glColor3f(1,0,0);



   BSAlgoLine(100,500,500,500);
   BSAlgoLine(100,500,200,200);
   BSAlgoLine(100,100,200,500);
   BSAlgoLine(500,500,200,500);

   BSAlgoLine(150,450,450,450);
   BSAlgoLine(150,450,250,250);
   BSAlgoLine(150,150,250,450);
   BSAlgoLine(450,450,250,450);

   BSAlgoLine(150,225,250,400);
   simpleLine(225,300,400,250);
   BSAlgoLine(300,375,250,400);
   simpleLine(375,450,400,250);

   BSAlgoLine(150,450,100,100);
   BSAlgoLine(150,200,100,200);
   simpleLine(400,450,200,100);
}

int main(int argc , char** argv){
   glutInit(&argc,argv);
   glutInitDisplayMode(GLUT_SINGLE);
   glutInitWindowPosition(0,0);
   glutInitWindowSize(600,600);
   glutCreateWindow("Object using Bresenham's Algorithm");
   glutDisplayFunc(primitive);
   glutMainLoop();
   return 0;
}
```

OUTPUT: