```
#include <GL/glut.h>
#include <stdlib.h>
#include<iostream>
                std;
void displayPoint(int x, int y)
{
glColor3f(0, 1, 0);
glPointSize(2);
glBegin(GL_POINTS);
glVertex2i(x, y);
glEnd();
}
float x01, x2, y01, y2;
int ch;
void SimpleLine(float x1, float y1, float x2, float y2)
{
float step;
float dx = x2 - x1;
float dy = y2 - y1;
   (abs(dx) >= abs(dy))
{
step = abs(dx);
}

step = abs(dy);
float Xinc = dx / (float)step;
float Yinc = dy / (float)step;
float x = x1;
float y = y1;
    (int i = 0; i <= step; i++)
{
displayPoint(x, y);
x = x + Xinc;
y = y + Yinc;
}
glFlush();
}

void DottedLine(float x1, float y1, float x2, float y2)
{
float step;
float dx = x2 - x1;
float dy = y2 - y1;
   (abs(dx) >abs(dy))
{
step = abs(dx);
}

step = abs(dy);
float Xinc = dx / (float)step;
float Yinc = dy / (float)step;
float x = x1;
float y = y1;
displayPoint(x, y);

    (int i = 0; i <= step; i++)
{
  (i%4==0)
{

displayPoint(x, y);
}
x = x + Xinc;
y = y + Yinc;
}
glFlush();
```

```
}
void DashedLine(float x1, float y1, float x2, float y2)
{
float step;
int count=0;
float dx = x2 - x1;
float dy = y2 - y1;
   (abs(dx) > abs(dy))
{
step = abs(dx);
}

step = abs(dy);

float Xinc = dx / (float)step;
float Yinc = dy / (float)step;
float x = x1;
float y = y1;

    (int i = 0; i <= step; i++)
{
count++;
   (count<7)
{
displayPoint(x, y);
x=x+Xinc;
y=y+Yinc;

}
       (count<=10 &&count>=7)
{
x=x+Xinc;
y=y+Yinc;
}
    {
x=x+Xinc;
y=y+Yinc;
count=0;
}
}
glFlush();
}
void SolidLine(float x1, float y1, float x2, float y2)
{
float step;
float dx = x2 - x1;
float dy = y2 - y1;
   (abs(dx) >= abs(dy))
{
step = abs(dx);
}

step = abs(dy);
float Xinc = dx / (float)step;
float Yinc = dy / (float)step;
float x = x1;

float y = y1;
    (int i = 0; i <= step; i++)
{
glColor3f(0, 1, 0);
glPointSize(5);
glBegin(GL_POINTS);
glVertex2i(x, y);
glEnd();
x = x + Xinc;
```

```
y = y + Yinc;
}
glFlush();
}
void myMouse(int button, int state, int x, int y)
{
static int xst, yst, pt = 0;
   (button == GLUT_LEFT_BUTTON &&state == GLUT_DOWN)
{
   (pt == 0)
{
xst = x;
yst = y;
x01 = xst;
y01 = yst;
pt = pt + 1;
}

{
x2 = x;
y2 = y;
   (ch == 1)
{
SimpleLine(xst, yst, x, y);
}
        (ch == 2)
{
DottedLine(xst, yst, x, y);
}
        (ch == 3)
{
DashedLine(xst, yst, x, y);
}
       (ch==4)
{
SolidLine(xst,yst,x,y);
}

xst = x;
yst = y;
}
}
        (button == GLUT_RIGHT_BUTTON && state == GLUT_DOWN)
pt = 0;
//Clear Screen
glFlush();
}
void keyboard(unsigned char key, int x, int y)
{
       (key)
{
     's':
ch = 1;
glutMouseFunc(myMouse);
     ;
     'd':
ch = 2;
glutMouseFunc(myMouse);
     ;
     'D':
ch = 3;
glutMouseFunc(myMouse);
     ;
     'S':
ch=4;
glutMouseFunc(myMouse);
```

```
     ;
}
glutPostRedisplay();
}
void initialize(void)
{
glClearColor(1.0, 1.0, 1.0, 1.0);
glClear(GL_COLOR_BUFFER_BIT);
// gluOrtho2D(l,r,b,t)
gluOrtho2D(0, 600, 600, 0);
}
void primitives(void)
{
//glClearColor(1.0, 1.0, 1.0, 1.0);

//glClear(GL_COLOR_BUFFER_BIT);
glColor3f(1, 0, 0);
SimpleLine(0, 300, 600, 300);
SimpleLine(300, 0, 300, 600);
glutKeyboardFunc(keyboard);
}
int main(int argc,char **argv)
{
glutInit(&argc, argv);
glutInitDisplayMode(GLUT_SINGLE);
glutInitWindowPosition(0, 0);
glutInitWindowSize(600, 600);
glutCreateWindow("OpenGL - DDA Algo");
initialize();
cout<<"--------------------";
cout<<"\ns. Simple Line";
cout<<"\nd. Dotted Line";
cout<<"\nD. Dashed Line";
cout<<"\nS. Solid Line";
cout<<"\n--------------------\n";
glutDisplayFunc(primitives);
glutMainLoop();
       0;
}
```