```
#include<GL/glut.h>
#include<stdlib.h>
#include<iostream>
                std;

void displayPoint(int x, int y)
{
        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 dx,dy,p;
        int i;
        float incx, incy, inc1, inc2;
        float x,y;

        dx= x2-x1;
        dy = y2-y1;

          (dx<0)
        {
                dx=-dx;
        }
          (dy<0)
        {
                dy=-dy;
        }
        incx=1;

          (x2<x1)
        {
                incx = -1;
        }
        incy=1;
          (y2<y1)
        {
                incy =-1;
        }
        x=x1;
        y=y1;

          (dx>dy)
        {
        glColor3f(0, 1, 1);
        displayPoint(x,y);
        p= 2*dy-dx;
        inc1 =2*(dy-dx);
        inc2 = 2*dy;

           (i=0; i<dx;i++)
        {
                  (p>=0)
                {
                        x= x+incx;
                        y=y+ incy;
                        p=p+inc1;
                }
```

```
                {
                        x=x+incx;
                        p=p+inc2;
                }
                glColor3f(0, 1, 1);
                displayPoint(x,y);
        }
        }

        {
                glColor3f(0, 1, 1);
                displayPoint(x,y);
                p= 2*dx-dy;
                inc1 = 2*(dx-dy);
                inc2 = 2*dx;
                   (i=0;i<dy;i++)
                {
                          (p>=0)
                        {
                                x=x+incx;
                                y=y+incy;
                                p=p+inc1;
                        }

                        {
                                p=p+inc2;
                                y=y+incy;
                        }
                        glColor3f(0, 1, 1);
                        displayPoint(x,y);
                }
        }
        glFlush();
}

void Dottedline(float x1, float y1, float x2, float y2)
{
        float dx,dy,p;
        int i;
        float incx, incy, inc1, inc2;
        float x,y;

        dx= x2-x1;
        dy = y2-y1;

          (dx<0)
        {
                dx=-dx;
        }
          (dy<0)
        {
                dy=-dy;
        }
        incx=1;

          (x2<x1)
        {
                incx = -1;
        }
        incy=1;
          (y2<y1)
        {
                incy =-1;
        }
        x=x1;
        y=y1;
```

```
  (dx>dy)
{
glColor3f(1, 0, 0);
displayPoint(x,y);
p= 2*dy-dx;
inc1 =2*(dy-dx);
inc2 = 2*dy;

   (i=0; i<dx;i++)
{

        (i%4==0)
        {
                  (p>=0)
                {
                        x= x+incx;
                        y=y+ incy;
                        p=p+inc1;
                }

                {
                        x=x+incx;
                        p=p+inc2;
                }
                glColor3f(1, 0, 0);
                displayPoint(x,y);
        }

        {
                  (p>=0)
                {
                        x= x+incx;
                        y=y+ incy;
                        p=p+inc1;
                }

                {
                        x=x+incx;
                        p=p+inc2;
                }
        }
  }

}

{
        glColor3f(1, 0, 0);
        displayPoint(x,y);
        p= 2*dx-dy;
        inc1 = 2*(dx-dy);
        inc2 = 2*dx;
           (i=0;i<dy;i++)
        {
                  (i%4==0)
                {
                          (p>=0)
                        {
                                x=x+incx;
                                y=y+incy;
                                p=p+inc1;
                        }

                        {
                                p=p+inc2;
                                y=y+incy;
```

```
                                }
                                glColor3f(1, 0, 0);
                                displayPoint(x,y);
                        }

                        {
                                (p>=0)
                                {
                                        x=x+incx;
                                        y=y+incy;
                                        p=p+inc1;
                                }

                                {
                                        p=p+inc2;
                                        y=y+incy;
                                }
                        }
                }
        }
        glFlush();
}


void Dashedline(float x1, float y1, float x2, float y2)
{
        float dx,dy,p;
        int i;
        float incx, incy, inc1, inc2;
        float x,y;
        int count=0;

        dx= x2-x1;
        dy = y2-y1;

          (dx<0)
        {
                dx=-dx;
        }
          (dy<0)
        {
                dy=-dy;
        }
        incx=1;

          (x2<x1)
        {
                incx = -1;
        }
        incy=1;
          (y2<y1)
        {
                incy =-1;
        }
        x=x1;
        y=y1;

          (dx>dy)
        {
        glColor3f(0,1,0);
        displayPoint(x,y);
        p= 2*dy-dx;
        inc1 =2*(dy-dx);
        inc2 = 2*dy;

           (i=0; i<dx;i++)
```

```
        {
                count++;
                  (count<=7)
                {

                          (p>=0)
                        {
                                x= x+incx;
                                y=y+ incy;
                                p=p+inc1;
                        }

                        {
                                x=x+incx;
                                p=p+inc2;
                        }
                        glColor3f(0,1,0);
                        displayPoint(x,y);
                }
                       (count<=10 && count>7)
                {
                          (p>=0)
                        {
                                x= x+incx;
                                y=y+ incy;
                                p=p+inc1;
                        }

                        {
                                x=x+incx;
                                p=p+inc2;
                        }
                }

                {
                        count =0;
                          (p>=0)
                        {
                                x= x+incx;
                                y=y+ incy;
                                p=p+inc1;
                        }

                        {
                                x=x+incx;
                                p=p+inc2;
                        }
                }
          }
        }

        {
                glColor3f(0,1,0);
                displayPoint(x,y);
                p= 2*dx-dy;
                inc1 = 2*(dx-dy);
                inc2 = 2*dx;

                   (i=0;i<dy;i++)
                {
                        count++;
                          (count<=7)
                        {

                                  (p>=0)
                                {
```

```
                                        x=x+incx;
                                        y=y+incy;
                                        p=p+inc1;
                                }

                                {
                                        p=p+inc2;
                                        y=y+incy;
                                }
                                glColor3f(0,1,0);
                                displayPoint(x,y);
                        }
                               (count<=10 && count>7)
                        {
                                  (p>=0)
                                {
                                        x=x+incx;
                                        y=y+incy;
                                        p=p+inc1;
                                }

                                {
                                        p=p+inc2;
                                        y=y+incy;
                                }
                        }

                        {
                                count=0;
                                  (p>=0)
                                {
                                        x=x+incx;
                                        y=y+incy;
                                        p=p+inc1;
                                }

                                {
                                        p=p+inc2;
                                        y=y+incy;
                                }
                        }
                }
        }
        glFlush();
}


void Solidline(float x1, float y1, float x2, float y2)
{
        float dx,dy,p;
        int i;
        float incx, incy, inc1, inc2;
        float x,y;

        dx= x2-x1;
        dy = y2-y1;

          (dx<0)
        {
                dx=-dx;
        }
          (dy<0)
        {
                dy=-dy;
        }
        incx=1;
```

```
  (x2<x1)
{
        incx = -1;
}
incy=1;
  (y2<y1)
{
        incy =-1;
}
x=x1;
y=y1;

  (dx>dy)
{
glPointSize(5);
glColor3f(1,0.5,0);
glBegin(GL_POINTS);
    glVertex2i(x,y);
glEnd();

p= 2*dy-dx;
inc1 =2*(dy-dx);
inc2 = 2*dy;

   (i=0; i<dx;i++)
{
          (p>=0)
        {
                x= x+incx;
                y=y+ incy;
                p=p+inc1;
        }

        {
                x=x+incx;
                p=p+inc2;
        }

        glPointSize(5);
        glColor3f(1,0.5,0);
        glBegin(GL_POINTS);
                glVertex2i(x,y);
        glEnd();
}
}

{
        glPointSize(5);
        glColor3f(1,0.5,0);
        glBegin(GL_POINTS);
                glVertex2i(x,y);
        glEnd();

        p= 2*dx-dy;
        inc1 = 2*(dx-dy);
        inc2 = 2*dx;
           (i=0;i<dy;i++)
        {
                  (p>=0)
                {
                        x=x+incx;
                        y=y+incy;
                        p=p+inc1;
                }
```

```
                        {
                                p=p+inc2;
                                y=y+incy;
                        }
                        glPointSize(5);
                        glColor3f(1,0.5,0);
                        glBegin(GL_POINTS);
                                glVertex2i(x,y);
                        glEnd();
                }
        }
        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;

        glFlush();
}


void Keyboard(unsigned char key, int x, int y)
{
               (key)
        {
                      's':
```

```
                        ch=1;
                        glutMouseFunc(Mymouse);
                             ;

                    'd':
                        ch=2;
                        glutMouseFunc(Mymouse);
                             ;

                    'a':
                        ch=3;
                        glutMouseFunc(Mymouse);
                             ;

                    'o':
                        ch=4;
                        glutMouseFunc(Mymouse);
                             ;

        }
        glutPostRedisplay();
}

void primitives(void)
{
        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);
        glutInitWindowSize(600,600);
        glutInitWindowPosition(0,0);
        glutCreateWindow("Bresenham's Algorithm - SIA03");
        glClearColor(1.0,1.0,1.0,1.0);
        glClear(GL_COLOR_BUFFER_BIT);
        gluOrtho2D(0.0, 599.0, 599.0, 0.0);

        cout<<"---------------------------------------";
        cout<<"\ns. Simple line";
        cout<<"\nd. Dotted line";
        cout<<"\na. Dashed line";
        cout<<"\no. solid line";
        cout<<"---------------------------------------";

        glutDisplayFunc(primitives);
        glutMainLoop();
               0;

}
```