```
/*Problem Statement :
Implement DDA and Bresenham line drawing algorithm to draw:
i) Simple Line
ii) Dotted Line
iii) Dashed
iv) Solid line
using mouse interface Divide the screen in four quadrants with
center as (0, 0). The line should work for all the slopes positive as well as negative.
NAME: Atharv Tambane
ROLL NO: S512078
*/

#include<windows.h>
#include<GL/glut.h>
#include<GL/glu.h>
#include<iostream>
#include<math.h>

#define h 700
#define w 700

using namespace std;

GLint xi, xii, yi, yii;

void setpixel(GLint x, GLint y)
{
glColor3f(0.0, 0.0, 1.0);
glBegin(GL_POINTS);
glVertex2f(x, y);
glEnd();
glFlush();
}

void initialize()
{
glClearColor(0.6, 0.6, 0.6, 0.0);
glClear(GL_COLOR_BUFFER_BIT);
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
gluOrtho2D(-w / 2, w / 2, -h / 2, h / 2);
}

void choice()
{
int i;
glPointSize(2.0);
for (i = -w; i < w; i++)
{
setpixel(0, i);
setpixel(i, 0);
}
}

class line
{
public:
void dda(int item)
{
GLfloat dx, dy, step, x, y;
GLfloat xinc, yinc;
int i;
dx = xii - xi;
dy = yii - yi;
if (abs(dx) >= abs(dy))
step = abs(dx);
else
```

```
 step = abs(dy);
 xinc = (float)dx / step;
 yinc = (float)dy / step;

 x = xi;
 y = yi;

 setpixel(x, y);
 for (i = 1; i <= step; i++)
 {
 x = x + xinc;
 y = y + yinc;
 xi = x + 0.5;
 yi = y + 0.5;
 if (item == 1)
 {
 setpixel(xi, yi);
 }
 if (item == 2)
 {
 if (i % 10 < 5)
{
 setpixel(xi, yi);
 }
 }
if (item == 3) { if (i % 9 >= 2 && i % 9 != 7) { setpixel(xi, yi); }
}
 if (item == 4)
 {
 glPointSize(4.0);
 setpixel(xi, yi);
 }
 }
 }

 void bresenham(int item)
 {
 int dx, dy, P, tmp;
 int i = 1;
 if (xii < xi && yii < yi)
 {
 tmp = xi;
 xi = xii;
 xii = tmp;
 tmp = yi;
 yi = yii;
 yii = tmp;
 }
 dx = (xii - xi);
 dy = (yii - yi);

 if (dy <= dx && dy > 0)
{
 dx = abs(dx);
 dy = abs(dy);
 P = (2 * dy) - dx;
 setpixel(xi, yi);
 int x = xi;
 int y = yi;
 while (x <= xii)
 {
 x++;
 if (P < 0)
 {
 P = P + (2 * dy);
 }
 else
```

```
 {
 y++;
 P = P + (2 * dy) - (2 * dx);
 }
 if (item == 1)
 {
 setpixel(x, y);
 }
 if (item == 2 && i % 10 < 5)
 {
 setpixel(x, y);
 }
if (item == 3 && (i % 9 >= 2 && i % 9 != 7))
 {
 setpixel(x, y);
 }
 if (item == 4)
 {
 glPointSize(4.0);
 setpixel(x, y);
 }
i++;
 }
 }
 else if (dy > dx && dy > 0)
 {
 dx = abs(dx);
 dy = abs(dy);
 P = (2 * dx) - dy;
 setpixel(xi, yi);
 int x = xi;
 int y = yi;
 while (y <= yii)
 {
 y++;
 if (P < 0)
 {
 P = P + (2 * dx);
 }
 else
 {
 x++;
P = P + (2 * dx) - (2 * dy);
 }
 if (item == 1)
 {
 setpixel(x, y);
 }
 if (item == 2 && i % 10 < 5)
 {
setpixel(x, y);
 }
if (item == 3 && (i % 9 >= 2 && i % 9 != 7))
 {
 setpixel(x, y);
 }
 if (item == 4)
 {
 glPointSize(4.0);
 setpixel(x, y);
 }
 i++;
 }
 }
 else if (dy >= -dx)
 {
 dx = abs(dx);
```

```
 dy = abs(dy);
 P = (2 * dy) - dx;
setpixel(xi, yi);
 int x = xi;
 int y = yi;
 while (x <= xii)
 {
 x++;
 if (P < 0)
 {
P = P + (2 * dy);
 }
 else
 {
y--;
 P = P + (2 * dy) - (2 * dx);
 }
 if (item == 1)
 {
 setpixel(x, y);
 }
 if (item == 2 && i % 10 < 5)
 {
 setpixel(x, y);
 }
 if (item == 3 && (i % 9 >= 2 && i % 9 != 7))
{
 setpixel(x, y);
 }
 if (item == 4)
 {
 glPointSize(4.0);
 setpixel(x, y);
 }
 i++;
 }
 }
 else if (dy < -dx)
{
 dx = abs(dx);
 dy = abs(dy);
 P = (2 * dy) - dx;
 setpixel(xi, yi);
 int x = xi;
 int y = yi;
 while (y >= yii)
 {
 y--;
 if (P < 0)
 {
 P = P + (2 * dx);
 }
 else
 {
 x++;
 P = P + (2 * dx) - (2 * dy);
 }
if (item == 1)
 {
 setpixel(x, y);
 }
 if (item == 2 && i % 10 < 5)
 {
 setpixel(x, y);
 }
 if (item == 3 && (i % 9 >= 2 && i % 9 != 7))
{
```

```
setpixel(x, y);
}
if (item == 4)
{
glPointSize(4.0);
setpixel(x, y);
}
i++;
}
}
glFlush();
}
};

line l;

void keyboard(unsigned char key, int x, int y)
{
if (key == 27)
exit(0);
else
cout << "You entered the " << key;
}

void menu(int item)
{
if (item == 1)
{
l.dda(1);
}
if (item == 2)
{
l.dda(2);
}
if (item == 3)
{
l.dda(3);
}
if (item == 4)
{
l.dda(4);
}
if (item == 5)
{
l.bresenham(1);
}
if (item == 6)
{
l.bresenham(2);
}
if (item == 7)
{
l.bresenham(3);
}
if (item == 8)
{
l.bresenham(4);
}
if (item == 9)
{
exit(0);
}
}

void mouse(int button, int state, int x, int y)
{
if (state == GLUT_DOWN)
```

```
 {
 if (button == GLUT_LEFT_BUTTON)
 {
 xi = x - w / 2;
 yi = h / 2 - y;
 cout << "Start point: (" << xi << ", " << yi << ")\n";
 }
 else if (button == GLUT_RIGHT_BUTTON)
{
 xii = x - w / 2;
 yii = h / 2 - y;
 cout << "End point: (" << xii << ", " << yii << ")\n";

 // Call the appropriate line drawing function
 // In this example, let's use DDA algorithm
l.dda(1); // 1 represents simple line in DDA function
}
}
 }

int main(int argc, char **argv)
{
 glutInit(&argc, argv);
 glutInitWindowSize(h, w);
 glutInitWindowPosition(100, 0);
 glutCreateWindow("Line DDA and Bresenham Atharv tamhane S5120078");
 initialize();
 glutDisplayFunc(choice);
 glutMouseFunc(mouse);
 glutKeyboardFunc(keyboard);
 glutCreateMenu(menu);
 glutAddMenuEntry("DDA_SIMPLE", 1);
 glutAddMenuEntry("DDA_DASH", 2);
 glutAddMenuEntry("DDA_DASH DOT", 3);
 glutAddMenuEntry("DDA_THICK", 4);
 glutAddMenuEntry("BRE_SIMPLE", 5);
 glutAddMenuEntry("BRE_DASH", 6);
 glutAddMenuEntry("BRE_DASH DOT", 7);
 glutAddMenuEntry("BRE_THICK", 8);
 glutAddMenuEntry("EXIT", 9);
 glutAttachMenu(GLUT_MIDDLE_BUTTON);
 glutMainLoop();
 return 0;
 }
```