```
#include<GL/gl.h>
#include<GL/glu.h>
#include<GL/glut.h>
#include<iostream>
using namespace std;
float x1, x2;
float y2, y3;
int ch;
void init(){
glClearColor(1,1,1,1);
glColor3f(0,0,0);
gluOrtho2D(-500,500,-500,500);
}
void display(){
float dy, dx, step, x, y, Xin, Yin;
glClear(GL_COLOR_BUFFER_BIT);
glPointSize(3);
glLineWidth(3);
dx = x2 - x1;
dy = y3 - y2;
if (abs(dx) > abs(dy))
{
step = abs(dx);
} else
{
step = abs(dy);
}
Xin = dx / step;
Yin = dy / step;
x = x1;
y = y2;
```

```
glBegin(GL_POINTS);
glVertex2i(x, y);
glEnd();
switch(ch){
int i;
case 1: {
for (i = 1; i <= step; i++) {
x = x + Xin;
y = y + Yin;
glBegin(GL_POINTS);
glVertex2i(x, y);
glEnd();
}
break;
}
case 2: {
for (i = 1; i <= step; i++) {
x = x + Xin;
y = y + Yin;
if(i%16<=8){
glBegin(GL_POINTS);
glVertex2i(x, y);
glEnd();
}
}
break;
}
case 3: {
for (i = 1; i <= step; i++) {
x = x + Xin;
y = y + Yin;
```

```
if(i%8==0){
glBegin(GL_POINTS);
glVertex2i(x, y);
glEnd();
}
}break;
}
case 4: {
for (i = 1; i <= step; i++) {
x = x + Xin;
y = y + Yin;
if(i%16<=4){
glBegin(GL_POINTS);
glVertex2i(x, y);
glEnd();
}
if(i%8==2){
glBegin(GL_POINTS);
glVertex2i(x, y);
glEnd();
}
}break;
}
}
glBegin(GL_LINES);
glVertex2i(-500,0);
glVertex2i(500,0);
glVertex2i(0,-500);
glVertex2i(0,500);
glEnd();
glFlush();
```

```
}
int main(int argc, char **argv){
cout<<"Enter x1 and y2"<<endl;
cin>>x1;
cin>>y2;
cout<<"Enter x2 and y3"<<endl;
cin>>x2>>y3;
cout<<"1.Simple 2.Dashed 3.Dotted 4.Center"<<endl;
cout<<"Enter your choice:"<<endl;
cin>>ch;
glutInit(&argc, argv);
glutInitDisplayMode(GLUT_RGB | GLUT_SINGLE);
glutInitWindowPosition(0,0);
glutInitWindowSize(500,500);
glutCreateWindow("DDA Line");
init();
glutDisplayFunc(display);
glutMainLoop();
return 0;
}
```