#if defined(\_WIN32) || defined(\_64)  
#include <windows.h>  
#endif  
#include <stdio.h>  
#include <math.h>  
#include <GL/glut.h>  
  
double X1, Y1, X2, Y2;  
  
float round\_value(float v)  
{  
    int z = v + 0.5;  
    return z;  
}  
void LineDDA(void)  
{  
  double dx=(X2-X1);  
  double dy=(Y2-Y1);  
  double steps;  
  float xInc,yInc,x=X1,y=Y1;  
  /\* Find out whether to increment x or y \*/  
  steps=(abs(dx)>abs(dy))?(abs(dx)):(abs(dy));  
  xInc=dx/(float)steps;  
  yInc=dy/(float)steps;  
  
  /\* Clears buffers to preset values \*/  
  glClear(GL\_COLOR\_BUFFER\_BIT);  
  
  /\* Plot the points \*/  
  glBegin(GL\_POINTS);  
  /\* Plot the first point \*/  
  glVertex2d(x,y);  
  int k;  
  /\* For every step, find an intermediate vertex \*/  
  for(k=0;k<steps;k++)  
  {  
    x+=xInc;  
    y+=yInc;  
    /\* printf("%0.6lf %0.6lf\n",floor(x), floor(y)); \*/  
    glVertex2d(round\_value(x), round\_value(y));  
  }  
  glEnd();  
  
  glFlush();  
}  
void Init()  
{  
  /\* Set clear color to white \*/  
  glClearColor(1.0,1.0,1.0,0);  
  /\* Set fill color to black \*/  
  glColor3f(0.0,0.0,0.0);  
  /\* glViewport(0 , 0 , 640 , 480); \*/  
  /\* glMatrixMode(GL\_PROJECTION); \*/  
  /\* glLoadIdentity(); \*/  
  gluOrtho2D(0 , 640 , 0 , 480);  
}  
int main(int argc, char \*\*argv)  
{  
  printf("Enter two end points of the line to be drawn:\n");  
  printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  
  printf("\nEnter Point1( X1 , Y1):\n");  
  scanf("%lf%lf",&X1,&Y1);  
  printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  
  printf("\nEnter Point1( X2 , Y2):\n");  
  scanf("%lf%lf",&X2,&Y2);  
  
  /\* Initialise GLUT library \*/  
  glutInit(&argc,argv);  
  /\* Set the initial display mode \*/  
  glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);  
  /\* Set the initial window position and size \*/  
  glutInitWindowPosition(0,0);  
  glutInitWindowSize(640,480);  
  /\* Create the window with title "DDA\_Line" \*/  
  glutCreateWindow("DDA\_Line");  
  /\* Initialize drawing colors \*/  
  Init();  
  /\* Call the displaying function \*/  
  glutDisplayFunc(LineDDA);  
  /\* Keep displaying untill the program is closed \*/  
  glutMainLoop();  
  return 0;  
}