#include<stdlib.h>  
#include <GL/glut.h>  
#include<math.h>  
  
int cnt;  
int sign(float x)  
{  
    if(x<0.0)  
        return -1 ;  
    else if (x>0.0)  
        return 1;  
    else  
        return 0 ;  
}  
  
void DDA(float x1,float y1,float x2,float y2)  
{  
    float length ;  
    if(abs(x2-x1)>=abs(y2-y1))  
    {  
        length=abs(x2-x1) ;  
    }  
    else  
    {  
        length=abs(y2-y1) ;  
    }  
    
    float delx,dely ;  
    delx=(x2-x1)/length ;  
    dely=(y2-y1)/length ;  
    float x=0.0,y=0.0 ;  
    x=x1+0.5\*sign(delx) ;  
    y=y1+0.5\*sign(dely) ;  
    int i=1 ;  
    while(i<=length)  
    {  
        glBegin(GL\_POINTS);  
        glColor3f(0.0, 1.0, 0.0);  
        glPointSize(16.0f) ;  
        glVertex2d((float)floor(x),(float)floor(y));  
        glEnd();  
        //glFlush ();  
        x=x+delx ;  
        y=y+dely ;  
        i++ ;  
        cnt++ ;  
    }  
  
}  
  
  
void Bcurve(float x1,float y1,float x2,float y2,float x3,float y3,float x4,float y4)  
{  
    float x,y,u,xd,yd ;  
    xd=x1;  
    yd=y1 ;  
    int i ;  
    for(i=0;i<1000;i++)  
    {  
        u=(float)i/1000 ;  
        x=x4\*pow(u,3)+3\*x3\*pow(u,2)\*(1-u)+3\*x2\*u\*pow((1-u),2)+x1\*pow((1-u),3) ;  
        y=y4\*pow(u,3)+3\*y3\*pow(u,2)\*(1-u)+3\*y2\*u\*pow((1-u),2)+y1\*pow((1-u),3) ;  
        glBegin(GL\_POINTS);  
        glColor3f(0.0, 0.0, 1.0);  
        glPointSize(8.0f) ;  
        glVertex2d((float)floor(x),(float)floor(y));  
        glEnd();  
  
    }  
      
}  
  
void Primitives(void)  
{  
//clear all pixels  
glClear (GL\_COLOR\_BUFFER\_BIT);  
  
//glBegin(GL\_LINES);  
//glVertex2d(0,0);  
//glVertex2d(200,300);  
//glVertex2d(400,300);  
//glVertex2d(350,487);  
//glVertex2d(450,487);  
//glVertex2d(500,400);  
//glEnd() ;  
cnt=0 ;  
int x=0 ;  
  
DDA(400,400,300,300) ;  
DDA(300,300,300,200) ;  
DDA(300,200,400,100) ;  
/\*  
DDA(450,100,700,300) ;  
DDA(700,300,600,300) ;  
DDA(600,300,600,500) ;  
DDA(600,500,300,500) ;  
x+=cnt ;  
cnt= 0 ;  
clipping(300,500,300,300);  
\*/  
  
  
Bcurve(400,400,300,300,300,200,400,100);  
  
  
glFlush ();  
}  
  
  
void Init()  
  
{  
  
glClearColor(1.0,1.0,1.0,0);  
  
    //glColor3f(1.0,0.0,0.0); // red  
  
//glViewport(0 , 0 , 800 , 800);  
glMatrixMode(GL\_PROJECTION);  
  
glLoadIdentity();  
  
gluOrtho2D(0 , 800 , 0 , 800);  
  
}  
  
int main(int argc, char \*\*argv)  
  
{  
  
glutInit(&argc,argv);  
  
glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);  
  
glutInitWindowSize(800,800);  
  
glutInitWindowPosition(80,80);  
  
glutCreateWindow("Primitives");  
  
Init();  
  
glutDisplayFunc(Primitives);  
  
glutMainLoop();  
  
return 0;  
  
}

