#include<GL/glut.h>  
#include <math.h>  
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 size)  
{  
    
    float delx,dely ;  
    delx=abs(x2-x1) ; dely=abs(y2-y1) ;  
    float x=0.0,y=0.0 ;  
    int s1=sign(x2-x1) ; int s2=sign(y2-y1) ;  
    x=x1 ;  
    y=y1 ;  
    int interchange=0 ;  
    if(dely>delx)  
    {  
        float temp=delx ;  
        delx=dely ;  
        dely=temp ;  
        interchange=1 ;  
    }  
    int i=1 ;  
    float e =2\*dely-delx ;  
    for(i=1;i<=delx;i++)  
    {  
         
            glPointSize(size) ;  
            glBegin(GL\_POINTS);  
            glColor3f(0.0, 1.0, 0.0);  
         
            glVertex2d((float)(x),(float)(y));  
            glEnd();  
        //glFlush ();  
        while(e>0)  
        {  
            if(interchange==1)  
                x=x+s1 ;  
            else  
                y=y+s2 ;  
            e=e-2\*delx ;  
        }  
        if(interchange==1)  
            y=y+s2 ;  
        else  
            x=x+s1 ;  
        e=e+2\*dely ;  
       // i++ ;  
    }  
  
}  
void circle\_fq(float h,float k,float R)  
{  
    float x1=0 ; float y1=R ;  
    float deli=2\*(1-R) ;  
    float del=0 ;  
    float deld=0 ;  
    while(y1>=0)  
    {  
        glPointSize(4.0) ;  
        glColor3f(0.0, 1.0, 0.0);  
        glBegin(GL\_POINTS);  
        glVertex2d((float)(x1+h),(float)(y1+k));  
        glVertex2d((float)(x1+h),(float)(k-y1));  
        glVertex2d((float)(h-x1),(float)(k-y1));  
        glVertex2d((float)(h-x1),(float)(y1+k));  
        glEnd();  
        if(deli<0)  
        {  
            del=2\*(deli+y1)-1 ;  
                if(del<=0)  
                {  
                    //mh  
                    x1=x1+1 ;  
                    deli=deli+2\*x1+1 ;  
                }  
                else  
                {  
                    //md  
                    x1=x1+1 ;  
                    y1=y1-1 ;  
                    deli=deli+2\*x1+2-2\*y1 ;  
                }  
        }  
        else if(deli>0)  
        {  
            deld=2\*deli-2\*x1-1 ;  
                if(deld<=0)  
                {  
                    //md  
                    x1=x1+1 ;  
                    y1=y1-1 ;  
                    deli=deli+2\*x1+2-2\*y1 ;  
                }  
                else  
                {  
                    //mv  
                    y1=y1-1 ;  
                    deli=deli+1-2\*y1 ;  
                  
                }  
  
        }  
        else if(deli==0)  
        {  
            x1=x1+1 ;  
            y1=y1-1 ;  
            deli=deli+2\*x1+2-2\*y1 ;  
        }  
  
    }  
  
  
}  
  
void elipse(float h,float k,float a,float b)  
{  
    float x1=0 ; float y1=b ;  
    float deli=a\*a-2\*a\*a\*b+b\*b ;  
    float del=0 ; float deld=0 ;  
    float a1=a ; float b1=b ;  
    while(y1>=0)  
    {  
        glPointSize(4.0) ;  
        glColor3f(0.0, 1.0, 0.0);  
        glBegin(GL\_POINTS);  
        glVertex2d((float)(x1+h),(float)(y1+k));  
        glVertex2d((float)(x1+h),(float)(k-y1));  
        glVertex2d((float)(h-x1),(float)(k-y1));  
        glVertex2d((float)(h-x1),(float)(y1+k));  
        glEnd();  
        if(deli<0)  
        {  
            del=2\*deli+2\*a\*a\*y1-a\*a ;  
                if(del<=0)  
                {  
                    //mh  
                    x1=x1+1 ;  
                    deli=deli+2\*b\*b\*(x1)+b\*b ;  
                }  
                else  
                {  
                    //md  
                    x1=x1+1 ;  
                    y1=y1-1 ;  
                    deli=deli+2\*b\*b\*x1-2\*a\*a\*y1+a\*a+b\*b ;  
                }  
        }  
        else if(deli>0)  
        {  
            deld=2\*deli-2\*b\*b\*x1-b\*b ;  
                if(deld<=0)  
                {  
                    //md  
                    x1=x1+1 ;  
                    y1=y1-1 ;  
                    deli=deli+2\*b\*b\*x1-2\*a\*a\*y1+a\*a+b\*b ;  
                }  
                else  
                {  
                    //mv  
                    y1=y1-1 ;  
                    deli=deli-2\*a\*a\*y1+a\*a ;  
                  
                }  
  
        }  
        else if(deli==0)  
        {  
            x1=x1+1 ;  
            y1=y1-1 ;  
            deli=deli+2\*b\*b\*x1-2\*a\*a\*y1+a\*a+b\*b ;  
        }  
  
    }  
  
}  
  
void Primitives(void)  
{  
//clear all pixels  
glClear (GL\_COLOR\_BUFFER\_BIT);  
circle\_fq(500,500,250);  
elipse(400,575,25,35);  
elipse(600,575,25,35);  
elipse(500,350,50,40);  
DDA(500,550,500,450,8.0) ;  
  
glFlush ();  
}  
void Init()  
  
{  
glClearColor(1.0,1.0,1.0,0);  
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;  
  
}

