**/\*WAP to implement Mid-Point Ellipse algorithm in Cpp.\*/**

**#include <iostream>//ellipse(x, y, start\_angle, end\_angle, x\_radius, y\_radius)**

**#include <cmath>**

**#include <graphics.h>**

**using namespace std;**

**int x\_c,y\_c;**

**float x\_n,y\_n,p\_1,p\_2,r\_x,r\_y;**

**void draw\_ellipse\_region\_1()**

**{**

**if (p\_1<0) //if p\_1<0 , x\_n = x\_n + 1, p\_1=p\_1+2\*r\_y^2\*x\_n+r\_y^2;**

**{**

**x\_n++; // change x\_n only**

**p\_1=p\_1+2\*r\_y\*r\_y\*x\_n+r\_y\*r\_y;**

**}**

**else //if 0<=p\_1 , x\_n = x\_n + 1, y\_n = y\_n - 1, p\_1=p\_1+2\*r\_y^2\*x\_n-2\*r\_x^2\*y\_n+r\_y^2**

**{**

**x\_n++; // change both x\_n and y\_n**

**y\_n--;**

**p\_1=p\_1+2\*r\_y\*r\_y\*x\_n-2\*r\_x\*r\_x\*y\_n+r\_y\*r\_y;**

**}**

**putpixel(x\_c+x\_n,y\_c+y\_n,GREEN); //1st quadrant**

**putpixel(x\_c-x\_n,y\_c+y\_n,GREEN); //2 nd quadrant**

**putpixel(x\_c-x\_n,y\_c-y\_n,GREEN); //3 rd quadrant**

**putpixel(x\_c+x\_n,y\_c-y\_n,GREEN); //4 th quadrant**

**}**

**void draw\_ellipse\_region\_2()**

**{**

**if (0<p\_2) // if 0<p\_2 , y\_n=y\_n-1, p\_2=p\_2-2\*r\_x^2\*y\_n+r\_x^2**

**{**

**y\_n--; // change y\_n only**

**p\_2=p\_2-2\*r\_x\*r\_x\*y\_n+r\_x\*r\_x;**

**}**

**else // if p\_2<=0 , y\_n=y\_n-1, x\_n = x\_n+1, p\_2=p\_2+2\*r\_y^2\*x\_n-2\*r\_x^2\*y\_n+r\_x^2**

**{**

**x\_n++; // change both x\_n & y\_n**

**y\_n--;**

**p\_2=p\_2+2\*r\_y\*r\_y\*x\_n-2\*r\_x\*r\_x\*y\_n+r\_x\*r\_x;**

**}**

**putpixel(x\_c+x\_n,y\_c+y\_n,GREEN); //1st quadrant**

**putpixel(x\_c-x\_n,y\_c+y\_n,GREEN); //2 nd quadrant**

**putpixel(x\_c-x\_n,y\_c-y\_n,GREEN); //3 rd quadrant**

**putpixel(x\_c+x\_n,y\_c-y\_n,GREEN); //4 th quadrant**

**}**

**int main()**

**{**

**int i;**

**while(1)**

**{**

**cout<<"\n\n\n\t\t\t\t\t1366\*768 ";**

**cout<<"\n\n\n\t\t Enter ellipse coordinates (x,y,r\_x,r\_y) with in range (0,0) to (1365,767)";**

**cout<<"\n\n Enter (x\_c,y\_c)";**

**cout<<"\n Enter x\_c: ";**

**cin>>x\_c;**

**cout<<" Enter y\_c: ";**

**cin>>y\_c;**

**cout<<"\n\n Enter r\_x: ";**

**cin>>r\_x;**

**cout<<" Enter r\_y: ";**

**cin>>r\_y;**

**x\_n=0;**

**y\_n=r\_y;**

**p\_1=r\_y\*r\_y-r\_x\*r\_x\*r\_y+r\_x\*r\_x/4; // p\_1=r\_y^2-r\_x^2\*r\_y+r\_x^2/4**

**initwindow(1366,768);**

**for(i=0; i<=1365; i++) // creates white background**

**{**

**line(0,i,1365,i);**

**}**

**//setcolor(GREEN);**

**//ellipse(x\_c+50,y\_c+50,0,360,r\_x,r\_y);**

**while ((r\_y\*r\_y\*x\_n)<=(r\_x\*r\_x\*y\_n)) //2\*r\_y^2\*x\_n<=2\*r\_x^2\*y\_n**

**{**

**draw\_ellipse\_region\_1();**

**}**

**p\_2=r\_y\*r\_y\*(x\_n+0.5)\*(x\_n+0.5)+r\_x\*r\_x\*(y\_n-1)\*(y\_n-1)-r\_x\*r\_x\*r\_y\*r\_y; //p\_2=r\_y^2\*(x\_n+0.5)^2+r\_x^2\*(y\_n-1)^2-r\_x^2\*r\_y^2;**

**while (0<y\_n)**

**{**

**draw\_ellipse\_region\_2();**

**}**

**putpixel(x\_c,y\_c,GREEN); //At center of ellipse**

**putpixel(x\_c,y\_c+r\_y,GREEN); //At topmost point**

**putpixel(x\_c,y\_c-r\_y,GREEN); //At bottom point**

**getch();**

**closegraph();**

**}**

**return 0;**

**}**

**/\*WAP to implement Mid-Point Ellipse algorithm in Cpp.\*/**

**//MID\_POINT\_Ellipse**

**//Using GLUT**

**#include<GL/gl.h>**

**#include<GL/glu.h>**

**#include<GL/glut.h>**

**//#include <bits/stdc++.h>**

**#include<iostream>**

**//for animation purpose**

**#include<vector>**

**using namespace std;**

**void display(); //display function**

**void reshape(int,int); //reshape the viewport**

**void timer(int); //for displaying no of frames in a sec**

**void getinfo(); //info from user**

**void drawEllipse(); // drawing circle**

**float xc,yc,a,b,p;**

**void drawEllipseAnimation(); //animation**

**void keyboard(unsigned char,int,int); //for animation keyboard input**

**float ax,ay,aa,ab,ap; //for animation points**

**bool startAnimation=false;//for animation start**

**vector<float> point;//for animation**

**bool once=false;**

**void init(){**

**glClearColor(0.1,0.1,0.1,1.0); //background color**

**}**

**int main(int argc, char\*\* argv){**

**getinfo();**

**glutInit(&argc,argv);**

**glutInitDisplayMode(GLUT\_RGB|GLUT\_DOUBLE);**

**glutInitWindowSize(500,500);**

**glutInitWindowPosition(200,200);**

**glutCreateWindow("Mid-Point-Ellipse");**

**glutDisplayFunc(display);**

**glutReshapeFunc(reshape);**

**glutSetKeyRepeat(GLUT\_KEY\_REPEAT\_OFF);**

**glutKeyboardFunc(keyboard);**

**glutTimerFunc(0,timer,0);**

**init();**

**glutMainLoop();**

**return 0;**

**}**

**void display(){**

**glClear(GL\_COLOR\_BUFFER\_BIT);**

**glLoadIdentity();**

**glColor3f(.7,.7,.7);//axis line color**

**glBegin(GL\_LINES);**

**glVertex2f(250,0);**

**glVertex2f(-250,0);**

**glVertex2f(0,250);**

**glVertex2f(0,-250);**

**glEnd();**

**glPointSize(3);**

**glBegin(GL\_POINTS);**

**glVertex2f(xc,yc);**

**glEnd();**

**glPointSize(1);**

**drawEllipse();**

**drawEllipseAnimation();**

**glutSwapBuffers();**

**}**

**void reshape(int w,int h){**

**glViewport(0,0,w,h);**

**glMatrixMode(GL\_PROJECTION);**

**glLoadIdentity;**

**gluOrtho2D(-250,250,-250,250);**

**glMatrixMode(GL\_MODELVIEW);**

**}**

**void timer(int){**

**glutPostRedisplay();**

**glutTimerFunc(1000/30,timer,0);**

**}**

**void getinfo(){**

**cout<<endl<<endl<<"\t Enter the following:"<<endl;**

**cout<<"\t Center x: ";**

**cin>>xc;**

**cout<<"\t Center y: ";**

**cin>>yc;**

**cout<<"\t a: ";**

**cin>>a;**

**cout<<"\t b: ";**

**cin>>b;**

**//for animation**

**ax=0;**

**ay=b;**

**aa=a;**

**ab=b;**

**ap=b\*b-a\*a\*b+(a\*a)/4;**

**}**

**void drawEllipse(){**

**float x,y;**

**p=b\*b-a\*a\*b+(a\*a)/4;**

**x=0;**

**y=b;**

**glColor3f(1,1,1);//ellipse color**

**glBegin(GL\_POINTS);**

**while(2\*b\*b\*x < 2\*a\*a\*y){**

**glVertex2f(xc+x,yc+y);**

**glVertex2f(xc+x,yc-y);**

**glVertex2f(xc-x,yc+y);**

**glVertex2f(xc-x,yc-y);**

**x=x+1;**

**if(p<0){**

**p=p+2\*b\*b\*x+b\*b;**

**}**

**else{**

**y=y-1;**

**p=p+2\*b\*b\*x+b\*b - 2\*a\*a\*y;**

**}**

**}**

**p=b\*b\*(x+.5)\*(x+.5) + a\*a\*(y-1)\*(y-1)-a\*a\*b\*b;**

**while(y>=0){**

**glVertex2f(xc+x,yc+y);**

**glVertex2f(xc+x,yc-y);**

**glVertex2f(xc-x,yc+y);**

**glVertex2f(xc-x,yc-y);**

**y=y-1;**

**if(p>0){**

**p=p-2\*a\*a\*y+a\*a;**

**}**

**else{**

**x=x+1;**

**p=p+2\*b\*b\*x+a\*a - 2\*a\*a\*y;**

**}**

**}**

**glEnd();**

**}**

**//For animation below here**

**void drawEllipseAnimation(){**

**if( 2\*ab\*ab\*ax < 2\*aa\*aa\*ay && startAnimation==true){**

**point.push\_back(xc+ax);**

**point.push\_back(yc+ay);**

**point.push\_back(xc+ax);**

**point.push\_back(yc-ay);**

**point.push\_back(xc-ax);**

**point.push\_back(yc+ay);**

**point.push\_back(xc-ax);**

**point.push\_back(yc-ay);**

**ax=ax+1;**

**if(ap<0){**

**ap=ap+2\*ab\*ab\*ax+ab\*ab;**

**}**

**else{**

**ay=ay-1;**

**ap=ap+2\*ab\*ab\*ax+ab\*ab - 2\*aa\*aa\*ay;**

**}**

**}**

**if(ay>=0 && 2\*ab\*ab\*ax > 2\*aa\*aa\*ay && startAnimation==true ){**

**if(!once){**

**ap=ab\*ab\*(ax+.5)\*(ax+.5) + aa\*aa\*(ay-1)\*(ay-1)-aa\*aa\*ab\*ab;**

**once=true;**

**}**

**point.push\_back(xc+ax);**

**point.push\_back(yc+ay);**

**point.push\_back(xc+ax);**

**point.push\_back(yc-ay);**

**point.push\_back(xc-ax);**

**point.push\_back(yc+ay);**

**point.push\_back(xc-ax);**

**point.push\_back(yc-ay);**

**ay=ay-1;**

**if(ap>0){**

**ap=ap-2\*aa\*aa\*ay+aa\*aa;**

**}**

**else{**

**ax=ax+1;**

**ap=ap+2\*ab\*ab\*ax+aa\*aa - 2\*aa\*aa\*ay;**

**}**

**}**

**if(ay>=0 || 2\*ab\*ab\*ax < 2\*aa\*aa\*ay)**

**glColor3f(1,0,0);**

**else**

**glColor3f(1,1,1);**

**glPointSize(1);**

**glBegin(GL\_POINTS);**

**for(int i=0;i<point.size();i+=2){**

**glVertex2f(point.at(i),point.at(i+1));**

**}**

**glEnd();**

**glPointSize(1);**

**}**

**void keyboard(unsigned char key,int x,int y){**

**if(key=='p')**

**startAnimation=true;**

**if(key=='o')**

**startAnimation=false;**

**}**