#include<iostream>

using namespace std;

#include<GL/glut.h>

int xc,yc,r,ch;

int point[4][2]={{300,300},{600,300},{600,600},{300,600}};

void myinit()

{

glClearColor(1.0,1.0,1.0,1.0);

glColor3f(0.0,1.0,0.0);

glPointSize(5.0);

gluOrtho2D(0.0,800,0.0,800);

}

void draw\_circle(int xc,int yc,int x,int y)

{

glBegin(GL\_POINTS);

glVertex2i(xc+x,yc+y);

glVertex2i(xc-x,yc+y);

glVertex2i(xc+x,yc-y);

glVertex2i(xc-x,yc-y);

glVertex2i(xc+y,yc+x);

glVertex2i(xc-y,yc+x);

glVertex2i(xc+y,yc-x);

glVertex2i(xc-y,yc-x);

glEnd();

}

void display\_cylinder(int xc,int yc,int r)

{

int x=0,y=r;

int d=3-2\*r;

while(x<y)

{

draw\_circle(xc,yc,x,y);

x++;

if(d<0)

d=d+4\*x+6;

else

{

y--;

d=d+4\*(x-y)+10;

}

draw\_circle(xc,yc,x,y);

}

}

void disply\_parallelopiped()

{

glBegin(GL\_LINE\_LOOP);

glVertex2i(point[0][0]--,point[0][1]--);

glVertex2i(point[1][0]--,point[1][1]--);

glVertex2i(point[2][0]--,point[2][1]--);

glVertex2i(point[3][0]--,point[3][1]--);

glEnd();

}

void draw\_cylinder()

{

int y,r1;

glClear(GL\_COLOR\_BUFFER\_BIT);

display\_cylinder(xc,yc,r);

glColor3f(1,0,0);

for(y=yc+1;y<300;y++)

display\_cylinder(xc,y,r);

glColor3f(0,0,0);

for(r1=0;r1<=r;r1++)

display\_cylinder(xc,y,r1);

glFlush();

}

void draw\_parallelopiped()

{

int y;

glClear(GL\_COLOR\_BUFFER\_BIT);

//back face of the parallelelopiped

glColor3f(0,0,1);

disply\_parallelopiped();

glColor3f(1,0,0);

for(y=100;y>0;y--)

disply\_parallelopiped();

glColor3f(0,0,1);

//front face of the parallelelopiped

glBegin(GL\_POLYGON);

glVertex2i(point[0][0]--,point[0][1]--);

glVertex2i(point[1][0]--,point[1][1]--);

glVertex2i(point[2][0]--,point[2][1]--);

glVertex2i(point[3][0]--,point[3][1]--);

glEnd();

glFlush();

}

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

{

cout<<"1:cylinder 2:parlelopiped\n";

cout<<"Enter your choice : \n";

cin>>ch;

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowSize(500,500);

glutInitWindowPosition(100,100);

if(ch==1)

{

cout<<"Enter circle coordinates : \n";

cin>>xc>>yc;

cout<<"Enter circle radius : \n";

cin>>r;

glutCreateWindow("Cylinder");

glutDisplayFunc(draw\_cylinder);

}

else

{

glutCreateWindow("Parallelopiped");

glutDisplayFunc(draw\_parallelopiped);

}

myinit();

glutMainLoop();

return 0;

}