**Program 11:**

#include<gl/glut.h>

#include<math.h>

#include<stdio.h>

#include "Header.h"

struct screenPt {

int x;

int y;

};

typedef enum { limacon = 1, cardioid = 2, threeLeaf = 3, spiral = 4 } curveName;

int w = 600, h = 500;

int curve = 1;

int red = 0, green = 0, blue = 0;

void myinit\_LabPA12(void) {

glClearColor(1.0, 1.0, 1.0, 1.0);

glMatrixMode(GL\_PROJECTION);

gluOrtho2D(0.0, 200.0, 0.0, 150.0);

}

void lineSegment(screenPt p1, screenPt p2) {

glBegin(GL\_LINES);

glVertex2i(p1.x, p1.y);

glVertex2i(p2.x, p2.y);

glEnd();

glFlush();

}

void drawCurve(int curveNum) {

const double twoPi = 6.283185;

const int a = 175, b = 60;

float r, theta, dtheta = 1.0 / float(a);

int x0 = 200, y0 = 250;

screenPt curvePt[2];

curve = curveNum;

glColor3f(red, green, blue);

curvePt[0].x = x0;

curvePt[0].y = y0;

glClear(GL\_COLOR\_BUFFER\_BIT);

switch (curveNum) {

case limacon: curvePt[0].x += a + b; break;

case cardioid: curvePt[0].x += a + a; break;

case threeLeaf: curvePt[0].x += a; break;

case spiral: break;

default: break;

}

theta = dtheta;

while (theta < twoPi) {

switch (curveNum) {

case limacon: r = a \* cos(theta) + b; break;

case cardioid: r = a \* (1 + cos(theta)); break;

case threeLeaf: r = a \* cos(3 \* theta); break;

case spiral: r = (a / 4.0) \* theta; break;

default: break;

}

curvePt[1].x = x0 + r \* cos(theta);

curvePt[1].y = y0 + r \* sin(theta);

lineSegment(curvePt[0], curvePt[1]);

curvePt[0].x = curvePt[1].x;

curvePt[0].y = curvePt[1].y;

theta += dtheta;

}

}

void colorMenu(int id) {

switch (id) {

case 0:

break;

case 1:

red = 0;

green = 0;

blue = 1;

break;

case 2:

red = 0;

green = 1;

blue = 0;

break;

case 4:

red = 1;

green = 0;

blue = 0;

break;

case 3:

red = 0;

green = 1;

blue = 1;

break;

case 5:

red = 1;

green = 0;

blue = 1;

break;

case 6:

red = 1;

green = 1;

blue = 0;

break;

case 7:

red = 1;

green = 1;

blue = 1;

break;

default:

break;

}

drawCurve(curve);

}

void main\_menu(int id) {

switch (id) {

case 3: exit(0);

default: break;

}

}

void mydisplay\_LabPA12() {

/\*int curveNum=1;

glClear(GL\_COLOR\_BUFFER\_BIT);

/\*printf("Enter the integer value corresponding to one of the followinf curve names:\n");

printf("1 - limacon\n2 - cardioid\n3 - threeleaf\n4 - spiral\n");

scanf\_s("%d", &curveNum);\*/

/\*if (curveNum == 1 || curveNum == 2 || curveNum == 3 || curveNum == 4)

drawCurve(curveNum);\*/

}

void myreshape(int nw, int nh) {

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(0.0, (double)nw, 0.0, (double)nh);

glClear(GL\_COLOR\_BUFFER\_BIT);

}

void LabPA\_12\_main(int argc, char\*\* argv) {

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(w, h);

glutInitWindowPosition(100, 100);

glutCreateWindow("Drawing curves");

int curveId = glutCreateMenu(drawCurve);

glutAddMenuEntry("Limacon", 1);

glutAddMenuEntry("Cardioid", 2);

glutAddMenuEntry("Threeleaf", 3);

glutAddMenuEntry("Spiral", 4);

glutAttachMenu(GLUT\_LEFT\_BUTTON);

int colorId = glutCreateMenu(colorMenu);

glutAddMenuEntry("Red", 4);

glutAddMenuEntry("Green", 2);

glutAddMenuEntry("Blue", 1);

glutAddMenuEntry("Black", 0);

glutAddMenuEntry("Yellow", 6);

glutAddMenuEntry("Cyan", 3);

glutAddMenuEntry("Magenta", 5);

glutAddMenuEntry("white", 7);

glutAttachMenu(GLUT\_LEFT\_BUTTON);

glutCreateMenu(main\_menu);

glutAddSubMenu("drawCurve", curveId);

glutAddSubMenu("colors", colorId);

glutAddMenuEntry("quit", 3);

glutAttachMenu(GLUT\_LEFT\_BUTTON);

myinit\_LabPA12();

glutDisplayFunc(mydisplay\_LabPA12);

glutReshapeFunc(myreshape);

glutMainLoop();

}

**OUTPUT**

