program 11

Create a menu with three Entries named curves, colors and quit. The Entry curves has a sub menu which has four Entries namely Limacon, cardiod, Three-leaf, and spiral. The color menu has sub menu with all Eight colors of RGB color model.

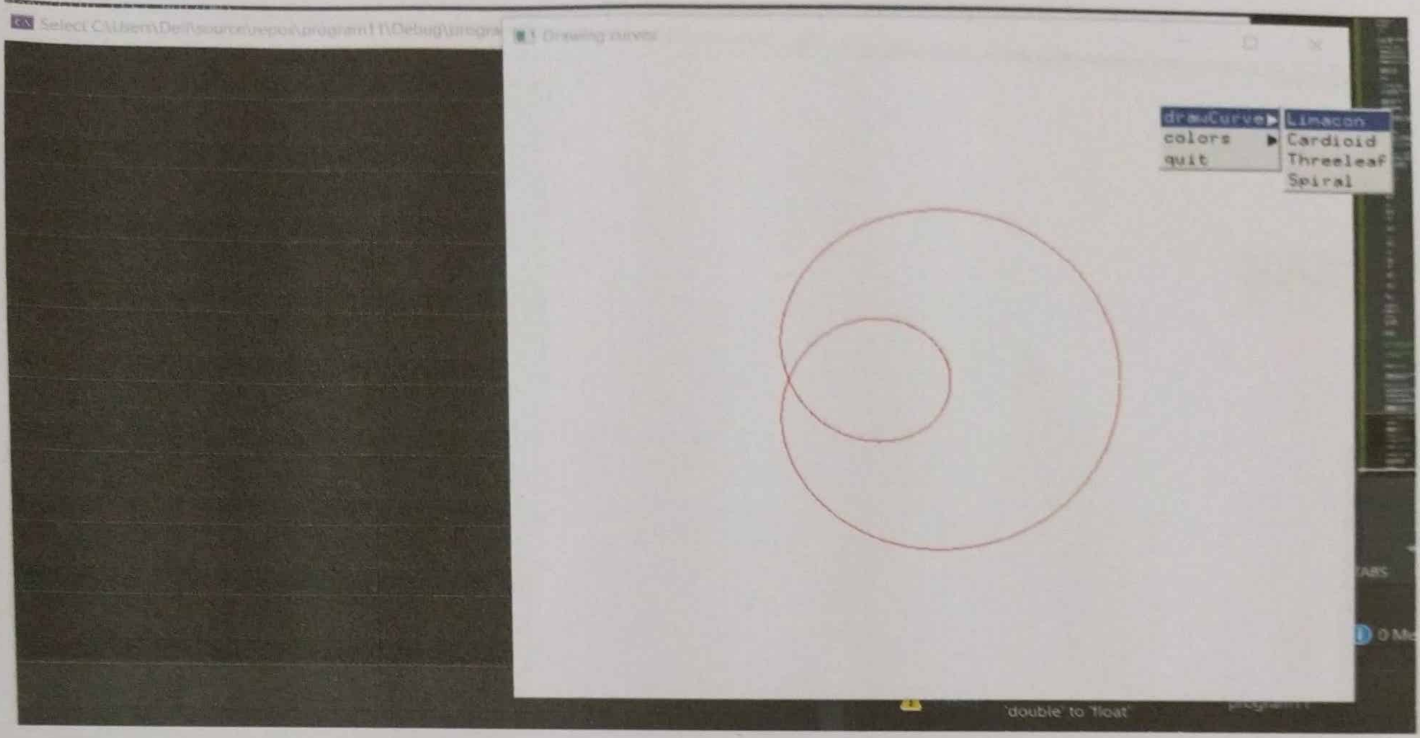
```
#include<gl/glut.h>
# include<math.h>
#include <stdio.h>
 struct screenPt{
        int x;
        int y;
 };
typedef  enum{ limacon=1, cardiod=2, threeleaf=3, spiral=4}
  curvename;

  int  w=600, h=500;
   int  curve=1;
    int  red=0, green=0, blue=0;
    void  myinit(void){
            glClearColor(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 ;     blue = 0;
                     green = 1;    break;
        case 3 :   red=0; green=0 ; blue=1; break;
        case 4 :   red=0;green=1; blue=0; 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 () {
        glClear (GL_COLOR_BUFFER_BIT);
        printf("1-limacon\n2-cardioid\n3-three leaf\n4-spiral\n");
        scanf("%d", &curvenum);
}
```

```
void main (int argc, char** argv){
        glutInit(&argc,argv);
        glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
        glutInitWindowSize(w,h);
        glutInitWindowPosition(100,100);
        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 (" white", 7);
          glutAttachMenu (GLUT_LEFTBUTTON);
          glutCreateMenu (main_Menu);
         glutAddSubMenu(" drawcurve", curveId);
          glutAddSubMenu ("colors", colorId);
         glutAddMenuEntry ("quit", 3);
         glutAttachMenu (GLUT_LEFT-BOTTON);
         myInit();
          glutDisplayFunc(mydisplay);
         glutMainLoop();
}
```

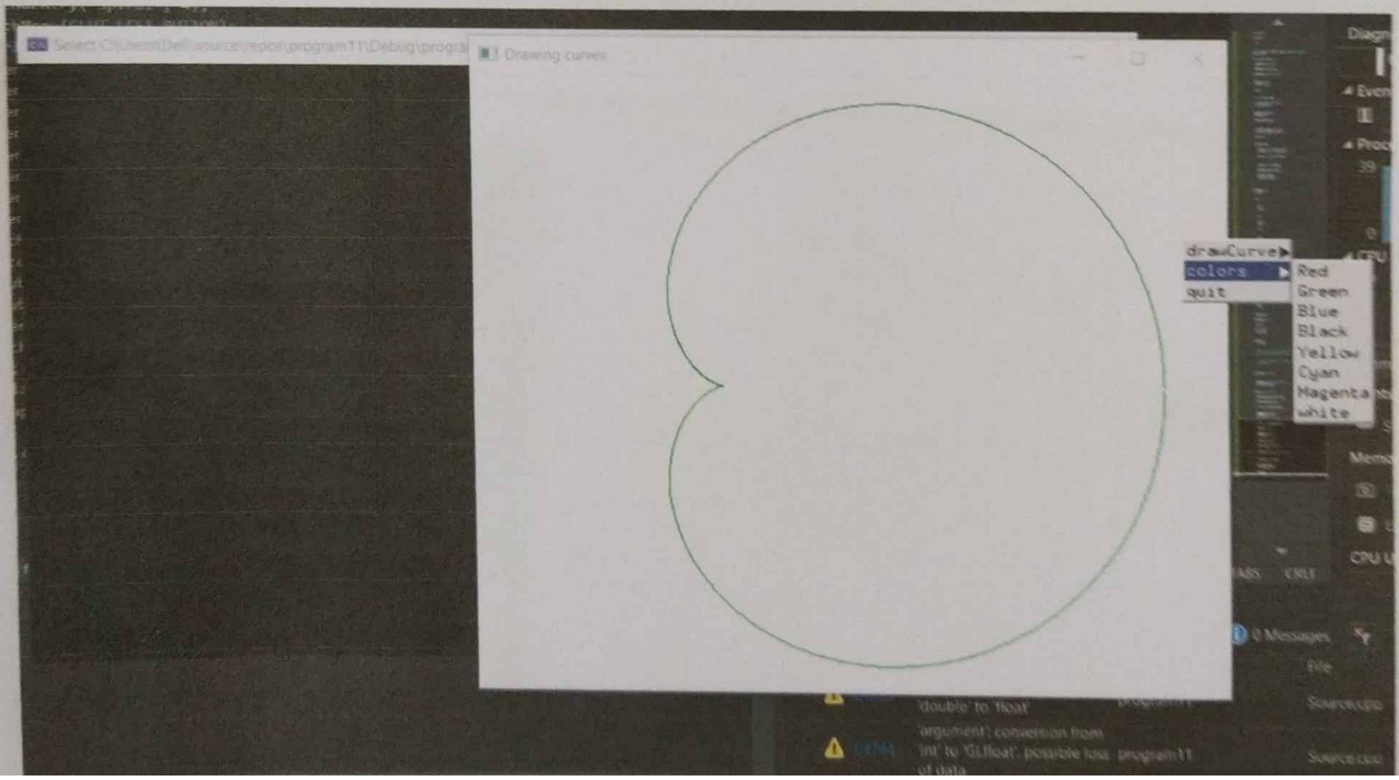
OUTPUT:



Select C:\Users\Dell\source\repos\program11\Debug\progra

Drawing curves

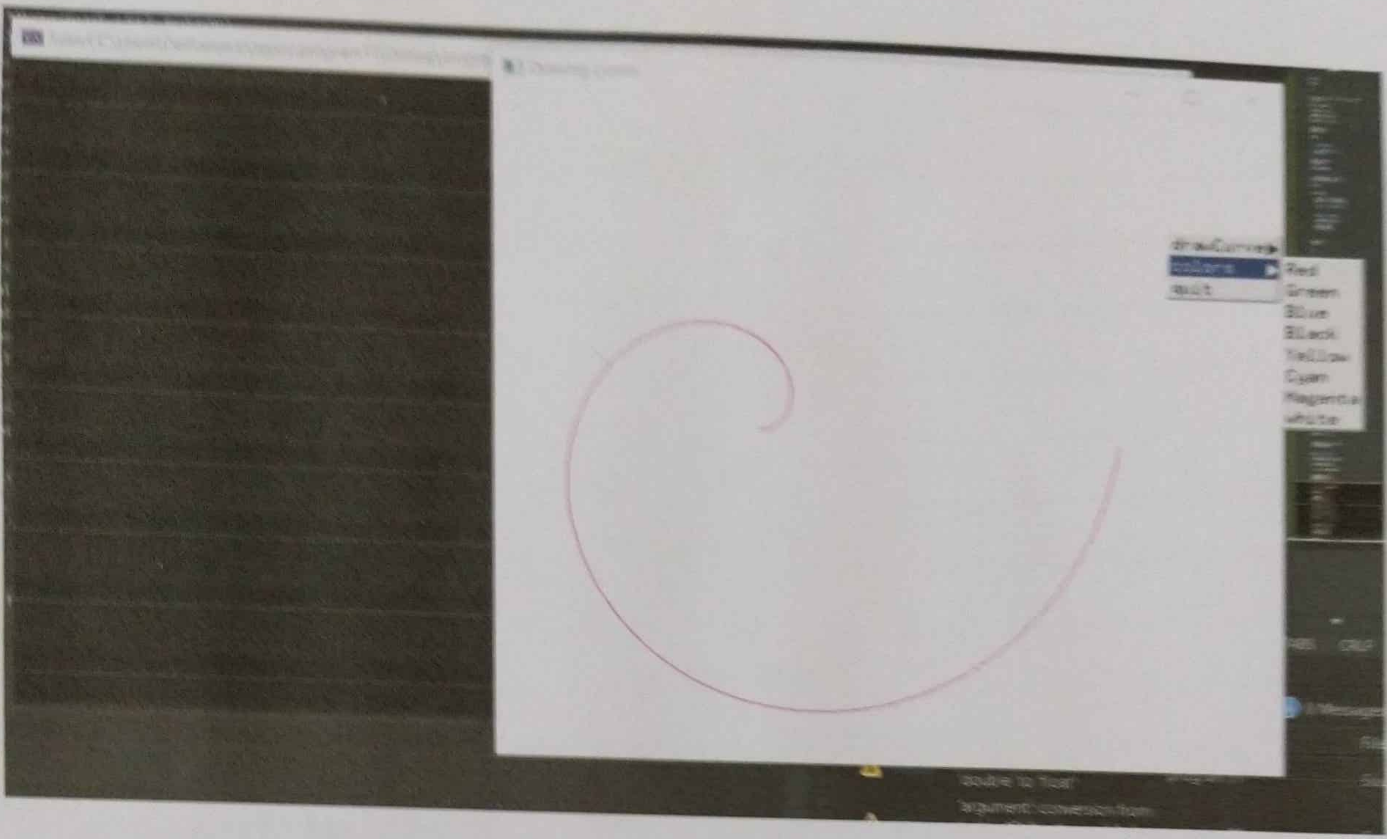
```
'double' to 'float'
```



Select C:\Users\Dell\source\repos\program11\Debug\progra

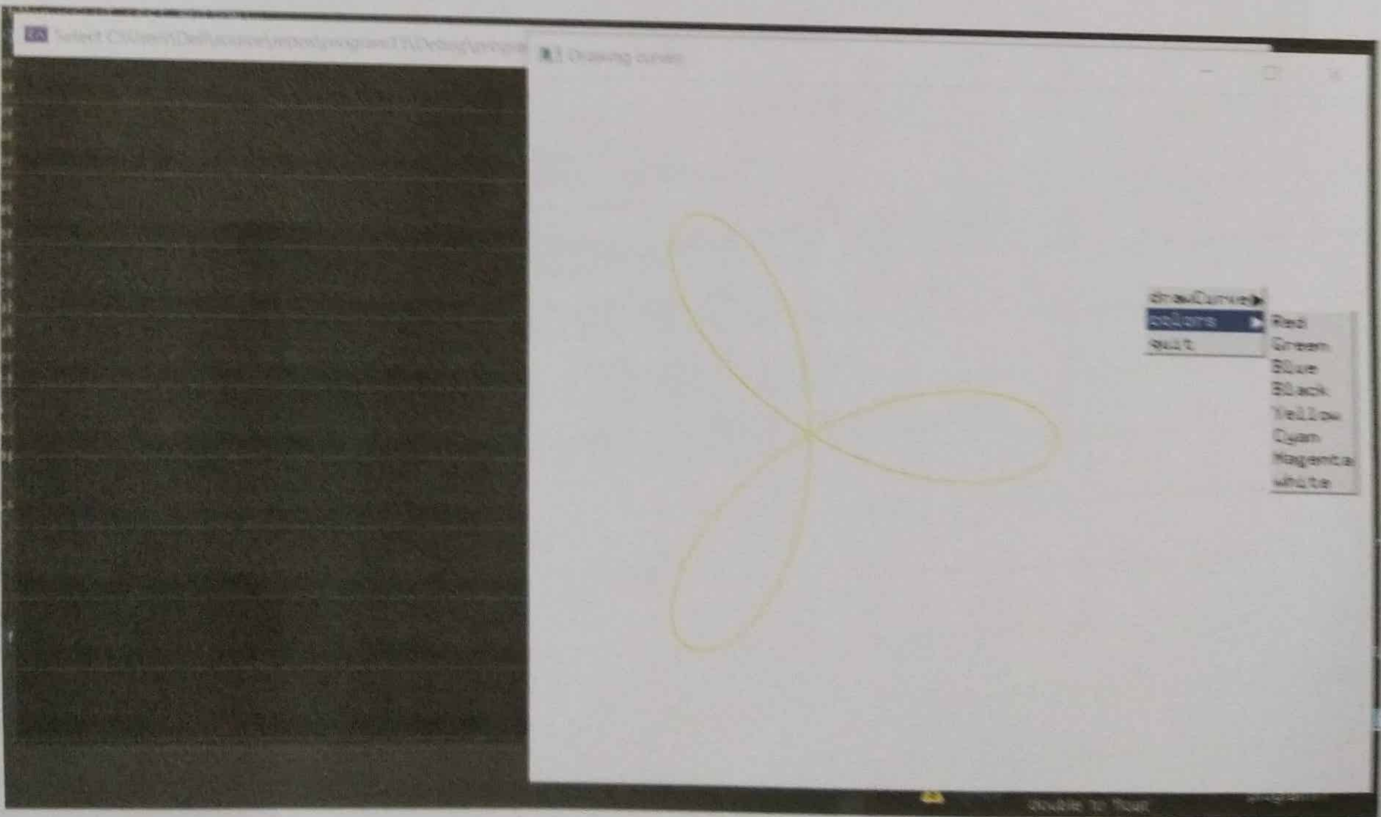
Drawing curves

```
'double' to 'float'                                          Source.cpp
'argument': conversion from
'int' to 'GLfloat', possible loss  program11                 Source.cpp
of data
```



double to 'float'

argument': conversion from



double to 'float'