

Aim:

Program to create a cylinder and a parallelepiped by extruding a circle and quadrilateral respectively.

###Algorithm:

1. Under ``Cylinder_Draw`` function, we call the ``Circle_draw`` function inside a for loop.
2. The ``Circle_Draw`` function is a midpoint circle drawing algorithm which draws a circle by calling ``plotpixels`` method. The ``plotpixels`` method draws the cylinder by plotting multiple pixels.
3. The ``parellopped_draw`` module calls the ``parellopped`` module inside a for loop which draws the parellelopiped onto the screen.

###Code:

```
//Cylinder and Parallelepiped by extruding Circle and
Quadrilateral
#include <GL/glut.h>
#include <math.h>
#include <stdio.h>
void draw_pixel(GLint cx, GLint cy)
{
    glColor3f(1.0,0.0,0.0);
    glBegin(GL_POINTS);
    glVertex2i(cx,cy);
    glEnd();
}
void plotpixels(GLint h, GLint k, GLint x, GLint y)
{
    draw_pixel(x+h,y+k);
    draw_pixel(-x+h,y+k);
    draw_pixel(x+h,-y+k);
    draw_pixel(-x+h,-y+k);
    draw_pixel(y+h,x+k);
    draw_pixel(-y+h,x+k);
    draw_pixel(y+h,-x+k);
    draw_pixel(-y+h,-x+k);
}
void Circle_draw(GLint h, GLint k, GLint r) // Midpoint Circle
Drawing Algorithm
{
    GLint d = 1-r, x=0, y=r;
    while(y > x)
    {
        plotpixels(h,k,x,y);
        if(d < 0)
            d+=2*x+3;
        else
        {
            d+=2*(x-y)+5;
            --y;
        }
    }
}
```

```

        ++x;
    }
    plotpixels(h,k,x,y);
}
void Cylinder_draw()
{
    GLint xc=100, yc=100, r=50;
    GLint i,n=50;
    for(i=0;i<n;i+=3)
    {
        Circle_draw(xc,yc+i,r);
    }
}
void parallelepiped(int x1,int x2,int y1, int y2, int y3, int
y4)
{
    glColor3f(0.0, 0.0, 1.0);
    glPointSize(2.0);
    glBegin(GL_LINE_LOOP);
    glVertex2i(x1,y1);
    glVertex2i(x2,y3);
    glVertex2i(x2,y4);
    glVertex2i(x1,y2);
    glEnd();
}
void parallelepiped_draw()
{
    int x1=200,x2=300,y1=100,y2=175,y3=100,y4=175;
    GLint i,n=40;
    for(i=0;i<n;i+=2)
    {
        parallelepiped(x1+i,x2+i,y1+i,y2+i,y3+i,y4+i);
    }
}
void init(void)
{
    to white    glColor(1.0,1.0,1.0,0.0); // Set display window color
    glMatrixMode(GL_PROJECTION); // Set Projection parameters
    gluOrtho2D(0.0,400.0,0.0,300.0);
}
void display(void)
{
    possible    glClear(GL_COLOR_BUFFER_BIT); // Clear Display Window
    glColor3f(1.0,0.0,0.0); // Set circle color to red (R G B)
    glPointSize(2.0);
    Cylinder_draw(); // Call cylinder
    parallelepiped_draw();// call parallelepiped
    glFlush(); // Process all OpenGL routines as quickly as
    possible
}
int main(int argc, char **argv)
{
    glutInit(&argc,argv); // Initialize GLUT

```

```

        glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB); // Set Display
mode      glutInitWindowPosition(50,50); // Set top left window
position  glutInitWindowSize(400,300); // Set Display window width and
height    glutCreateWindow("Cylinder and parallelepiped Display by
Extruding Circle and Quadrilateral "); // Create Display Window
        init();
        glutDisplayFunc(display); // Send the graphics to Display
Window    glutMainLoop();
        return 0;
    }

```

Output:

Commands for execution:-

* Open a terminal and Change directory to the file location in both the terminals.

* compile as gcc -lGLU -lGL -lglut parallelopiped.c -o parallelopiped

* If no errors, run as ./parallelopiped

Screenshots:-

![Screenshot of Output](parallelopiped.png)