Experiment – 8

Aim:

To print characters using Stroke generation method

Theory:

Stroke character generation method is a technique used to create characters or letters by drawing lines between specified points. In this method, a set of points or coordinates are defined to create the basic shape of each character. The lines connecting these points are then drawn to create the final character.

Stroke character generation method is widely used in computer graphics, especially in applications that require the generation of text, such as word processors, graphic design software, and video games.

The `draw_r()`, `draw_a()`, `draw_h()`, `draw_u()`, and `draw_l()` functions each define a series of vertices using the `glBegin()` and `glEnd()` functions to generate the corresponding stroke for the letters 'r', 'a', 'h', 'u', and 'l'.

The `display()` function calls each of these stroke-generating functions in sequence to generate the entire name "rahul". The `glClear()` and `glFlush()` functions are used to clear the color buffer and render the strokes to the screen, respectively.

The `main()` function initializes GLUT, sets up the window and the projection, and registers the `display()` function as the callback function to be executed when the window is displayed. Finally, the `glutMainLoop()` function starts the main event processing loop, which handles user input and updates the display as needed.

Implementation:

```
#include <GL/glut.h>
void draw_r()
{
   glBegin(GL_LINE_STRIP);
   glVertex2f(-0.8, 0.8);
   glVertex2f(-0.8, -0.8);
   glVertex2f(-0.4, -0.8);
   glVertex2f(-0.4, 0.4);
   glVertex2f(-0.8, 0.4);
   glEnd();
}
void draw_a()
{
```

```
    glBegin(GL_LINE_STRIP);
    glVertex2f(-0.2, 0.8);
    glVertex2f(0.2, 0.8);
    glVertex2f(0.6, -0.8);
    glVertex2f(0.2, 0.4);
    glVertex2f(-0.2, 0.4);
    glEnd();

    glBegin(GL_LINE_STRIP);
    glVertex2f(0.2, 0.4);
    glVertex2f(-0.2, 0.4);
    glEnd();
}

void draw_h()
{
    glBegin(GL_LINE_STRIP);
    glVertex2f(0.8, 0.8);
    glVertex2f(0.8, -0.8);
    glEnd();

    glBegin(GL_LINE_STRIP);
    glVertex2f(1.2, 0.8);
    glVertex2f(1.2, -0.8);
    glEnd();

    glBegin(GL_LINE_STRIP);
    glVertex2f(0.8, 0);
    glVertex2f(1.2, 0);
    glEnd();
}

void draw_u()
```

```
{
   glBegin(GL_LINE_STRIP);
   glVertex2f(1.6, 0.8);
   glVertex2f(1.6, -0.4);
   glVertex2f(2, -0.8);
   glVertex2f(2.4, -0.4);
   glVertex2f(2.4, 0.8);
   glEnd();
}

void draw_l()
{
   glBegin(GL_LINE_STRIP);
   glVertex2f(2.8, 0.8);
   glVertex2f(2.8, -0.8);
   glVertex2f(3.2, -0.8);
   glEnd();
}

void display()
{
   glClear(GL_COLOR_BUFFER_BIT);
   glColor3f(1.0, 1.0, 1.0);
   draw_r();
   draw_a();
   draw_h();
   draw_u();
   draw_l();
   glFlush();
}

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

```
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(500, 500);
    glutInitWindowPosition(100, 100);
    glutCreateWindow("Stroke Generation for Rahul");
    glClearColor(0.0, 0.0, 0.0, 0.0);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(-3.0, 3.0, -3.0, 3.0);
    glutDisplayFunc(display);
    glutMainLoop();
    return 0;
}
```