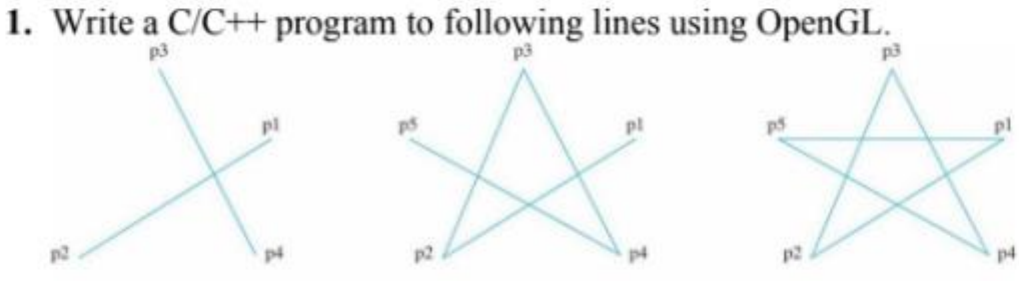
CGV Practical : 3



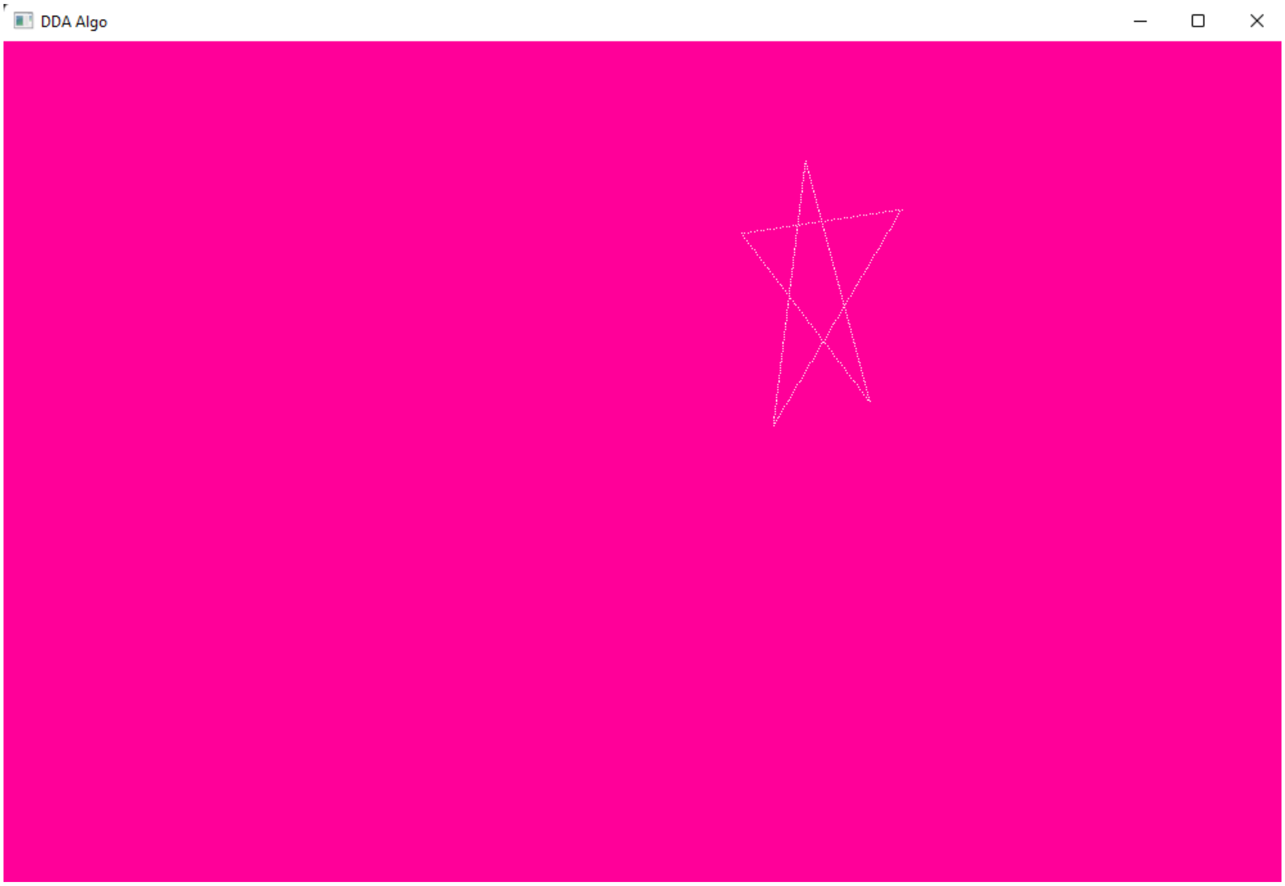
1. Write a C/C++ program to following lines using OpenGL.

Code :

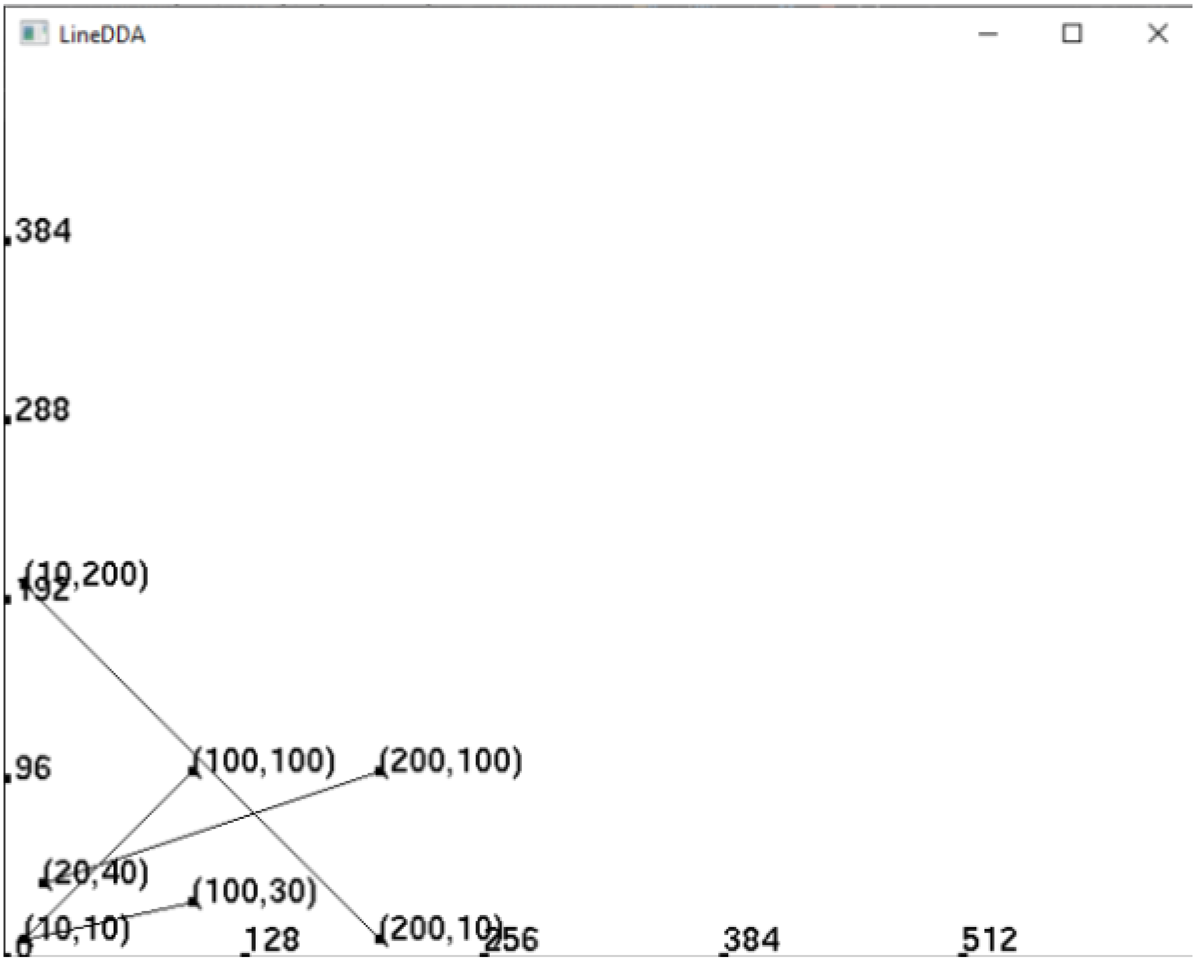
```
namespace pgp31
{
    void setPixel(double x, double y)
    {
        glBegin(GL_POINTS);
        glVertex2d(x, y);
        glEnd();
    }
    void dda(float x1, float x2, float y1, float y2) // call when window need to re-drwan
    {
        float dx = x2 - x1, dy = y2 - y1;
        float xInc = 0, yInc = 0, x = x1, y = y1, steps = 0;
        steps = (fabs(dx) > fabs(dy)) ? fabs(dx) : fabs(dy);
        //steps = ((dx) > (dy)) ? (dx) : (dy);
        xInc = dx / (float)steps;
        yInc = dy / (float)steps;
        setPixel(x, y);
        for (int k = 0; k < steps; k++)
        {
            x += xInc;
            y += yInc;
            setPixel(x, y);
        }
        setPixel(x, y);
        glFlush();//display framebuffer on screen

    }
    void main()  // for clear color
    {
        glClear(GL_COLOR_BUFFER_BIT);
        dda(10, 20, 30, 40);
        dda(10, 120, 60, 40);
        glFlush();
```

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

OUTPUT :



2. Write a C/C++ program to draw a line using DDA line drawing algorithm.
   Example: Line1(10,10), (100,100)
   Line2 (10,10), (100,30)
   Line3 (200,100), (20,40)
   Line4 (10,200), (200,10)



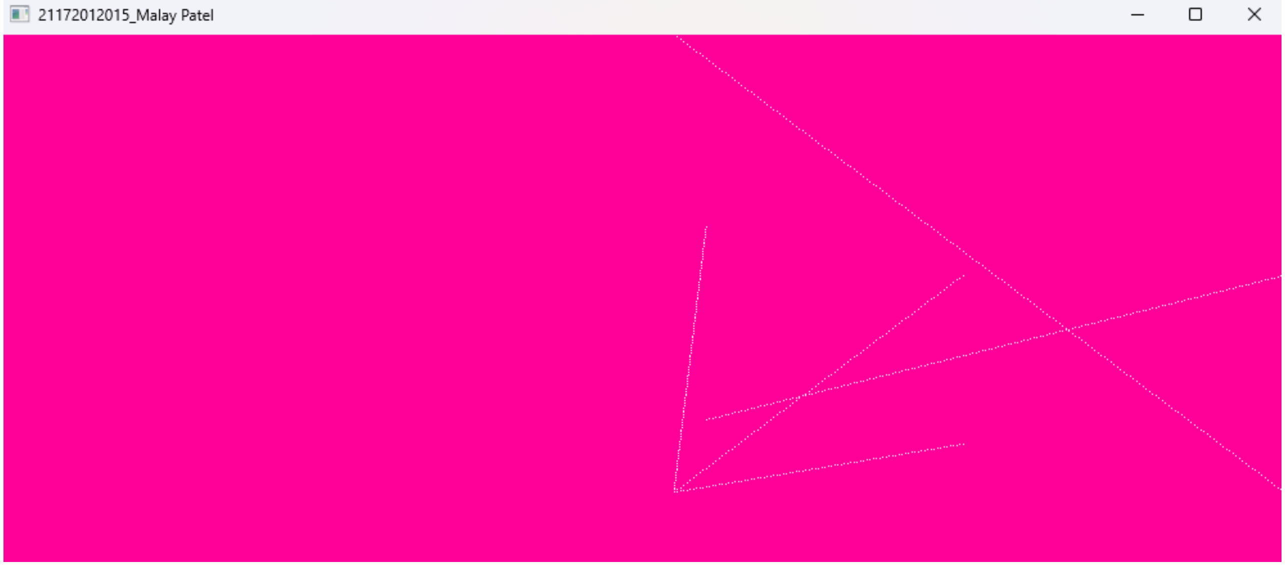
Code :

```
namespace gp32
{
    void setPixel(double x, double y)
    {
        glBegin(GL_POINTS);
        glVertex2d(x, y);
        glEnd();
    }
    void dda(float x1, float y1, float x2, float y2)
    {
        float dx = x2 - x1, dy = y2 - y1;
        float xInc = 0, yInc = 0, x = x1, y = y1, steps = 0;
        steps = (fabs(dx) > fabs(dy)) ? fabs(dx) : fabs(dy);
        xInc = dx / (float)steps;
        yInc = dy / (float)steps;
        setPixel(x, y);
        for (int k = 0; k < steps; k++)
        {
            x += xInc;
            y += yInc;
            setPixel(x, y);
        }
        setPixel(x, y);
    }
    void display()
    {
        glClear(GL_COLOR_BUFFER_BIT);
        dda(10, 10, 100, 100);
        dda(10, 10, 100, 30);
        dda(200, 100, 20, 40);
        dda(10, 200, 200, 10);
        dda(10, 10, 20, 120);
        //glPointSize(5.0);
        /*glBegin(GL_POINTS);
        glVertex2d(10, 20);
        glVertex2d(30, 40);
        glVertex2d(10, 120);
        glVertex2d(60, 40);
        glEnd();*/
        /**Draw a red x - axis, a green y - axis, and a blue z - axis.Each of
        the
         // axes are ten units long.
        glBegin(GL_LINES);
        glColor3f(1, 0, 0); glVertex3f(0, 0, 0); glVertex3f(100, 0, 0);
        glColor3f(0, 1, 0); glVertex3f(0, 0, 0); glVertex3f(0, 100, 0);
        glColor3f(0, 0, 1); glVertex3f(0, 0, 0); glVertex3f(0, 0, 100);
        glEnd();*/
        //glColor3f(1.0, 1.0, 1.0);
        //glutWireTorus(0.5, 3, 15, 30);
        // Draw the tetrahedron. It is a four sided figure, so when defining it
            // with a triangle strip we have to repeat the last two vertices.
            glFlush();
    }
    void reshape(int, int);
```

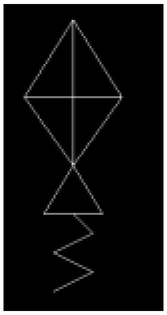
AB3

```
    void init() // for clear color
    {
        glClearColor(1.0, 0.0, 0.6, 1.0); //lies between 0-1 for color intensity
    }
    void main(int argc, char** argv) // command line arguments
    {
        glutInit(&argc, argv); // initialized glut library
        glutInitDisplayMode(GLUT_RGB); // Display mode that glut will use
        glutInitWindowPosition(100, 100); // create window with windows.
        glutInitWindowSize(1000, 1000); //width & height of window or size of
window
            glutCreateWindow("21172012015_Malay Patel");//with title of
window..1 order
            glutDisplayFunc(display); //2 order
        glutReshapeFunc(reshape);
        init();
        glutMainLoop(); // loop run continues to display windows
    }
    void reshape(int w, int h)//resize clipping area
    {
        glViewport(0, 0, (GLsizei)w, (GLsizei)h); //everything’s draw inside it
            glMatrixMode(GL_PROJECTION);//change mode or rotation or scaling
        glLoadIdentity();//reset all parameters
        gluOrtho2D(-200, 200, -200, 200);
        glMatrixMode(GL_MODELVIEW);//change mode
    }
}
```

OUTPUT :



3. Write a C/C++ Program to draw kite as given in image.

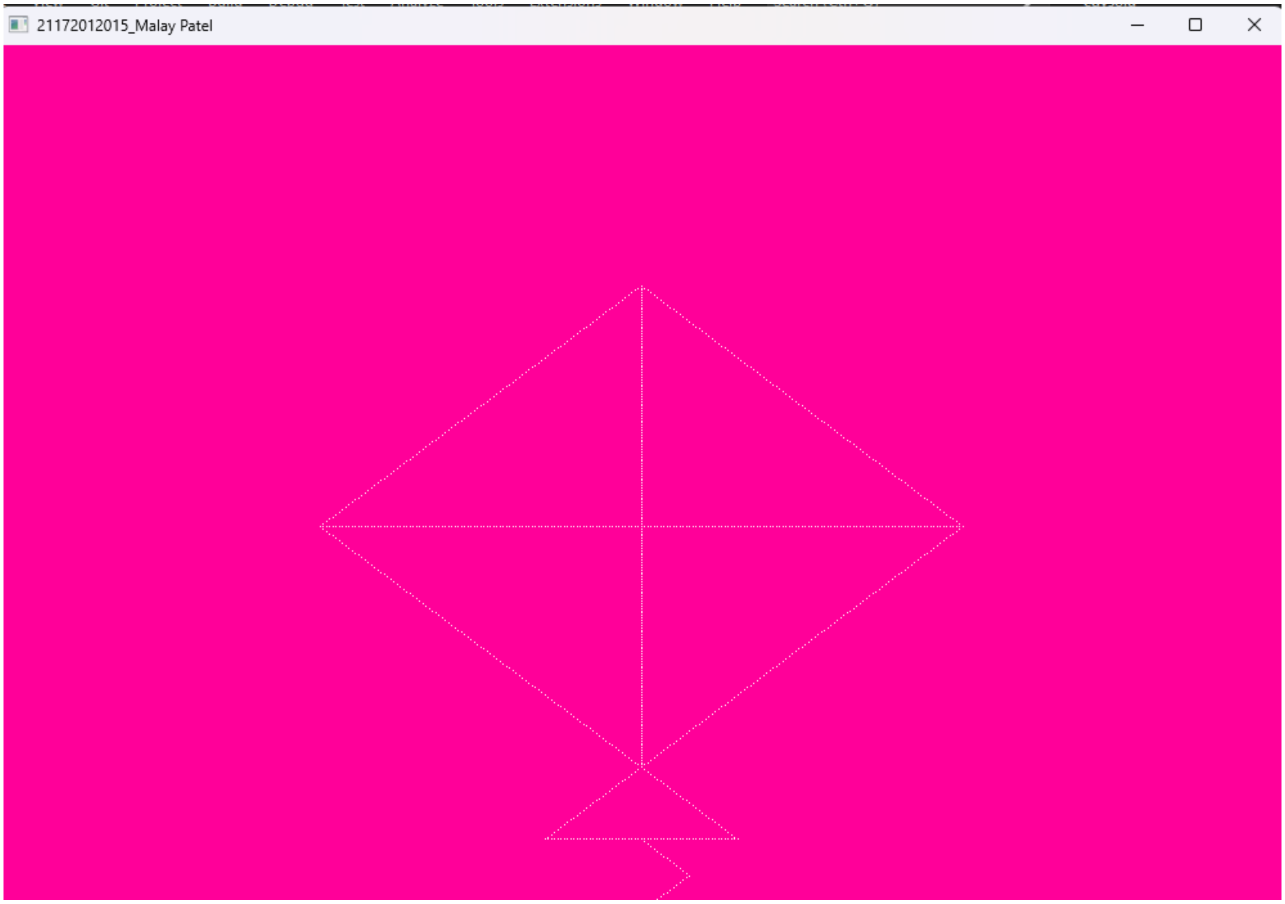


Code :

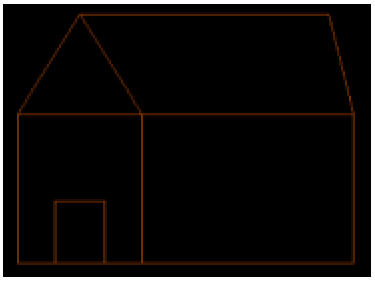
```
namespace pgp33 {

	void setPixel(double x, double y)
	{
		glBegin(GL_POINTS);
		glVertex2d(x, y);
		glEnd();
	}
	void dda(float x1, float y1, float x2, float y2)
	{
		float dx = x2 - x1, dy = y2 - y1;
		float xInc = 0, yInc = 0, x = x1, y = y1, steps = 0;
		steps = (fabs(dx) > fabs(dy)) ? fabs(dx) : fabs(dy);
		xInc = dx / (float)steps;
		yInc = dy / (float)steps;
		setPixel(x, y);
		for (int k = 0; k < steps; k++)
		{
			x += xInc;
			y += yInc;
			setPixel(x, y);
		}
		setPixel(x, y);
	}
	void display()
	{
		glClear(GL_COLOR_BUFFER_BIT);
		dda(100, 0, -100, 0);
		dda(0, 100, 0, -100);
		dda(100, 0, 0, 100);
		dda(0, 100, 100, 0);
		dda(-100, 0, 0, -100);
		dda(0, -100, -100, 0);
		dda(0, 100, -100, 0);
		dda(100, 0, 0, -100);
```

```
        dda(0, -100, -30, -130);
        dda(0, -100, 30, -130);
        dda(-30, -130, 30, -130);
        dda(0, -130, 15, -145);
        dda(15, -145, 0, -160);
        dda(0, -160, 15, -175);
        //glPointSize(5.0);
        /*glBegin(GL_POINTS);
        glVertex2d(10, 20);
        glVertex2d(30, 40);
        glVertex2d(10, 120);
        glVertex2d(60, 40);
        glEnd();*/
        /**Draw a red x - axis, a green y - axis, and a blue z - axis.Each of
        the
        // axes are ten units long.
        glBegin(GL_LINES);
        glColor3f(1, 0, 0); glVertex3f(0, 0, 0); glVertex3f(100, 0, 0);
        glColor3f(0, 1, 0); glVertex3f(0, 0, 0); glVertex3f(0, 100, 0);
        glColor3f(0, 0, 1); glVertex3f(0, 0, 0); glVertex3f(0, 0, 100);
        glEnd();*/
        //glColor3f(1.0, 1.0, 1.0);
        //glutWireTorus(0.5, 3, 15, 30);
        // Draw the tetrahedron. It is a four sided figure, so when defining it
            // with a triangle strip we have to repeat the last two vertices.
            glFlush();
    }
    void reshape(int, int);
    void init() // for clear color
    {
        glClearColor(1.0, 0.0, 0.6, 1.0); //lies between 0-1 for color intensity
    }
    void main(int argc, char** argv) // command line arguments
    {
        glutInit(&argc, argv); // initialized glut library
        glutInitDisplayMode(GLUT_RGB); // Display mode that glut will use
        glutInitWindowPosition(100, 100); // create window with windows.
        glutInitWindowSize(1000, 1000); //width & height of window or size of
window
            glutCreateWindow("21172012015_Malay Patel");//with title of
window..1 order
            glutDisplayFunc(display); //2 order
        glutReshapeFunc(reshape);
        init();
        glutMainLoop(); // loop run continues to display windows
    }
    void reshape(int w, int h)//resize clipping area
    {
        glViewport(0, 0, (GLsizei)w, (GLsizei)h); //everything’s draw inside it
        glMatrixMode(GL_PROJECTION);//change mode or rotation or scaling
        glLoadIdentity();//reset all parameters
        gluOrtho2D(-200, 200, -200, 200);
        glMatrixMode(GL_MODELVIEW);//change mode
    }
}
```

OUTPUT :



4. Write a C/C++ Program to draw hut as given in image

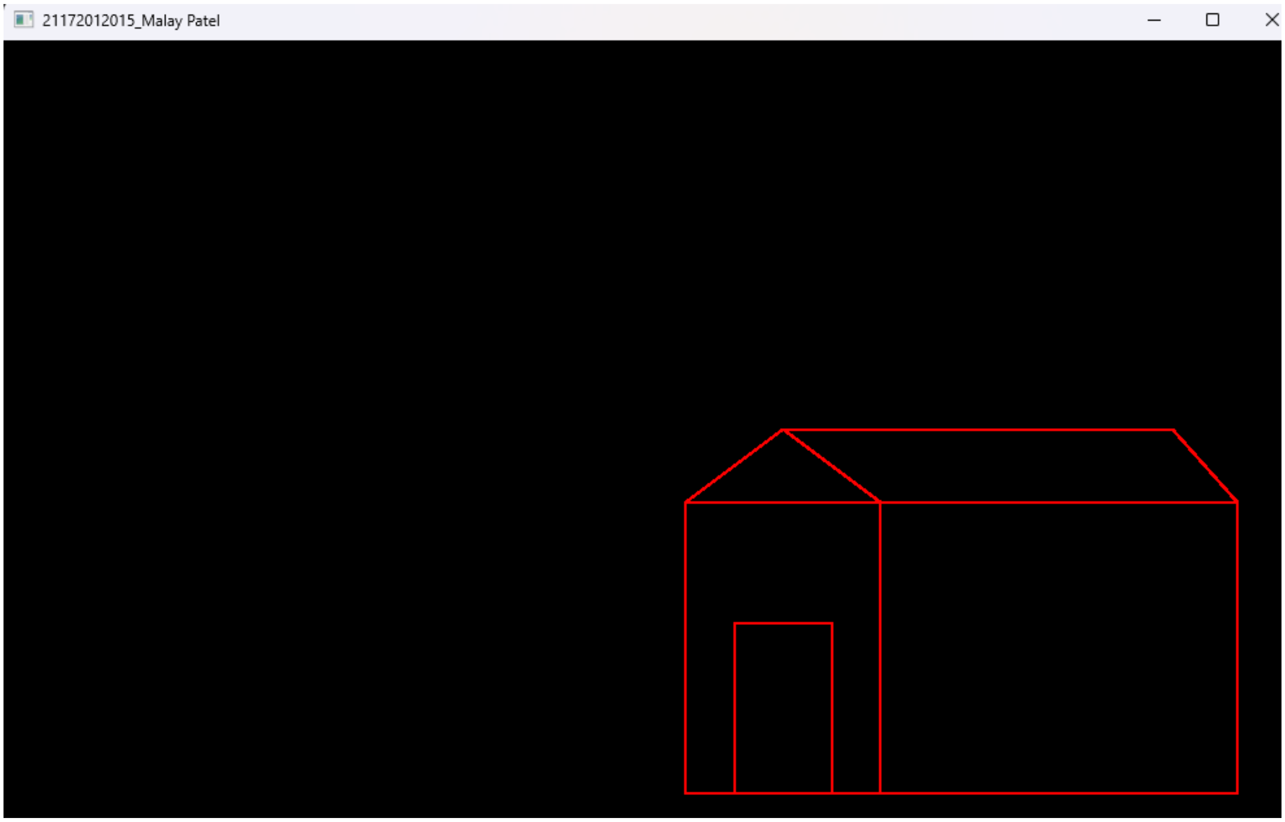


Code :

```
namespace gp34
{
    void setPixel(double x, double y)
    {
        glBegin(GL_POINTS);
        glVertex2d(x, y);
        glEnd();
    }

    void dda(float x1, float y1, float x2, float y2)
    {
        float dx = x2 - x1, dy = y2 - y1;
        float xInc = 0, yInc = 0, x = x1, y = y1, steps = 0;
        steps = (fabs(dx) > fabs(dy)) ? fabs(dx) : fabs(dy);
        xInc = dx / (float)steps;
        yInc = dy / (float)steps;
        setPixel(x, y);
        for (int k = 0; k < steps; k++)
        {
            x += xInc;
            y += yInc;
            setPixel(x, y);
        }
        setPixel(x, y);
    }
    void display()
    {
        glClear(GL_COLOR_BUFFER_BIT);
        glColor3f(1, 0, 0);
        glPointSize(2);
        dda(50, 50, 200, 200); //1
        dda(200, 200, 800, 200); //2
        dda(800, 200, 900, 50);//3
        dda(900, 50, 50, 50);//4
        dda(200, 200, 350, 50);//5
        dda(50, 50, 50, -550);//6
        dda(50, -550, 900, -550);//7
        dda(125, -550, 125, -200);//8
        dda(125, -200, 275, -200);//9
        dda(275, -200, 275, -550);//10
        dda(350, 50, 350, -550);//11
        dda(900, -550, 900, 50);//12
        glFlush();
    }
    void reshape(int, int);
    void init()  // for clear color
    {
        glClearColor(0.0, 0.0, 0.0, 0.0);    //lies between 0-1  for color
intensity
    }
    void main(int argc, char** argv)      // command line arguments
    {
        glutInit(&argc, argv);        // initialized glut library
        glutInitDisplayMode(GLUT_RGB); // Display mode that glut will use
        glutInitWindowPosition(100, 100);  // create window with windows.
```

```
		glutInitWindowSize(1000, 1000);  //width & height of window or size of
window
		glutCreateWindow("21172012015_Malay Patel");//with title of window..1
order
		glutDisplayFunc(display);         //2 order
		glutReshapeFunc(reshape);
		init();
		glutMainLoop();     // loop run continues to display window
	}
	void reshape(int w, int h)//resize clipping area
	{
		glViewport(0, 0, (GLsizei)w, (GLsizei)h);//everything’s draw inside it

		glMatrixMode(GL_PROJECTION);//change mode or rotation or scaling
		glLoadIdentity();//reset all parameters
		gluOrtho2D(-1000, 1000, -1000, 1000);
		glMatrixMode(GL_MODELVIEW);//change mode
	}

}
```

OUTPUT :


21172012015_Malay Patel