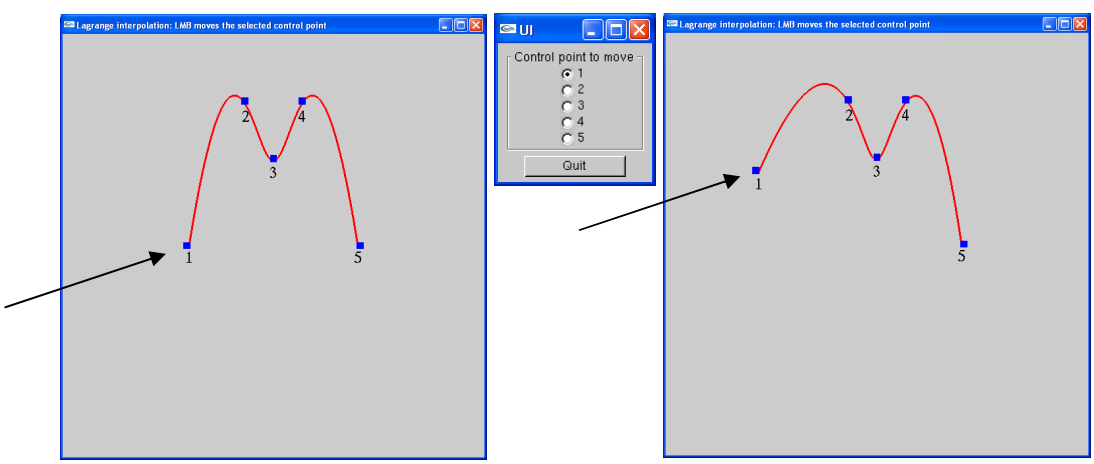
Complete the given source code so as to implement an OpenGL/GLUT program that uses parametric Lagrange interpolation to draw a curve through 5 points in the XY plane ($n + 1 = 5$). The points are given: $P_0 = (x_0, y_0), P_1, P_2, P_3, P_4$.

$$Q(t) = P_0 L_0^4(t) + P_1 L_1^4(t) + \cdots + P_4 L_4^4(t), \text{ where}$$

$$L_i^4(t) = \prod_{j=0}^{4} \frac{(t - t_j)}{(t_i - t_j)}, (j \neq i) = \frac{(t - t_0)(t - t_1)(t - t_2)(t - t_3)(t - t_4)}{(t_i - t_0)(t_i - t_1)(t_i - t_2)(t_i - t_3)(t_i - t_4)}, (j \neq i)$$

- Implement the function **`draw_curve()`**:
  - For values of the parameter t, $0 \le t \le 1$, compute points (x, y) on the curve. Draw the curve as a GL_LINE_STRIP between these points.
  - You will evaluate $Q(t)$ twice, once using the x-coordinates of the control points, $x = Q_x(t)$, and once using the y-coordinates, $y = Q_y(t)$.
- To compute $Q(t) = P_0 L_0^4(t) + P_1 L_1^4(t) + \cdots + P_4 L_4^4(t)$, implement the function **`Lin(i,t)`** that returns $L_i^4(t)$. Use equally spaced values for t_k:

  $t_0 = \frac{0}{4}, t_1 = \frac{1}{4}, t_2 = \frac{2}{4}, t_3 = \frac{3}{4}, t_4 = \frac{4}{4}$

Note: User-interface functionality is already implemented that lets you move control points: After selecting a control point to move, a left mouse button click will re-position that control point to the new location.