**Algorithm for 2D transformation**

 **User Input:**

* The code starts by prompting the user to input the coordinates of two points: (x1, y1) and (x2, y2). These represent the starting and ending points of the line.

 **Bresenham's Line Drawing Algorithm:**

* This part of the code draws a line between two points (x0, y0) and (x1, y1) using Bresenham's algorithm, which efficiently determines the pixels to plot in order to form a straight line.
* **Steps in Bresenham's algorithm:**
  + Calculate the differences dx and dy between the x and y coordinates.
  + Initialize a decision parameter p and begin iterating along the x-axis.
  + If p is negative, the next point is chosen along the x-axis. If p is positive or zero, the next point moves diagonally (i.e., in both x and y directions).
  + This loop continues until the x-coordinate reaches x1.

 **2D Transformation (Scaling and Translation):**

* Once the line is drawn, the code applies a series of transformations to the points.
* **Scaling:**
  + A scaling matrix is used to scale the line in the x-direction by a factor of 4 and in the y-direction by a factor of 1.5.
* **Translation:**
  + A translation matrix shifts the line by translating it by (x1, y1) to move it to a new position.
* The **inverse translation matrix** is then used to undo the translation, effectively shifting the points back.

 **Composite Transformation:**

* The scaling, translation, and inverse translation are combined into a **composite matrix**. This matrix is applied to the original coordinates of the line to get the transformed coordinates.
* The transformed line is then plotted along with the original line.

 **Plotting:**

* The code uses **matplotlib** to display the original line (in blue) and the transformed line (in red) on a graph.