***Pseudocode***

***1. Initialization:***

* Declare variables x1, y1, x2, y2, distance, slope, and y\_intercept
* Set distance, slope, and y\_intercept to 0 (initial values)

***2. Input Coordinates:***

* Print a message prompting the user to enter the first point's coordinates.
* Read the x-coordinate of the first point and store it in x1.
* Read the y-coordinate of the first point and store it in y1.
* Print a message prompting the user to enter the second point's coordinates.
* Read the x-coordinate of the second point and store it in x2.
* Read the y-coordinate of the second point and store it in y2.

***3. Calculate Distance:***

* Define a function functDistance
  + Inside the function:
    - Calculate the distance between the two points using the distance formula: distance = sqrt((x2 - x1)\*\*2 + (y2 - y1)\*\*2) (square root of the squared differences in x and y coordinates)
    - Set the distance variable to the calculated value.
    - Print a message displaying the calculated distance.

***4. Calculate Slope:***

* Define a function functSlope
  + Inside the function:
    - Calculate the slope of the line using the slope formula: slope = (y2 - y1) / (x2 - x1) (change in y divided by change in x)
    - Set the slope variable to the calculated value.
    - Print a message displaying the calculated slope.

***5. Calculate Equation:***

* Define a function functEquation
  + Inside the function:
    - Calculate the slope again (same formula as before).
    - Calculate the y-intercept using the point-slope form: y\_intercept = y1 - slope \* x1 (y-coordinate of one point minus slope times x-coordinate of the same point)
    - Set the y\_intercept variable to the calculated value.
    - Print a message displaying the equation of the line in the form y = mx + b (where m is slope and b is y-intercept).