Exercise 9  (10 points)

Execute the ICP algorithm to align the set of 2D points $P$ to those of set $Q$ (as close as possible).

$$P = \left\{ \left( \begin{array}{c} 1 \\ 2 \\ 3 \end{array} \right), \left( \begin{array}{c} 1.5 \\ 3.8 \\ 5 \end{array} \right), \left( \begin{array}{c} 2.7 \\ 6 \\ 8 \end{array} \right), \left( \begin{array}{c} 2.5 \\ 7 \\ 8 \end{array} \right), \left( \begin{array}{c} 3 \\ 8 \\ 8.5 \end{array} \right) \right\}$$

$$Q = \left\{ \left( \begin{array}{c} 2 \\ 3 \\ 3 \end{array} \right), \left( \begin{array}{c} 3 \\ 4 \\ 3 \end{array} \right), \left( \begin{array}{c} 4 \\ 5 \\ 4 \end{array} \right), \left( \begin{array}{c} 5 \\ 5 \\ 5 \end{array} \right), \left( \begin{array}{c} 5 \\ 6 \\ 6 \end{array} \right) \right\}$$

Also, compare between using the closest point and point-to-line matching methods to find the corresponding points of $Q$ and $P$.

Note: The line is represented by the closest point and the closer of its two direct neighbours.

Exercise Steps:

Implement the missing parts in 'src/09_icp/src/ICP.cpp' according to the following instructions:

a) Implement the function `distance` that calculates the Euclidean distance between a pair of 2D points.

b) Implement the function `closestPointOnLine` that computes the closest point that lies on a given line to a given 2D point.

c) Implement the function `min` that gets the minimum value within a vector of values.

d) Implement the function `euclideanCorrespondences` that computes the corresponding points in $P$ to those points in $Q$, using the closest point matching method.
e) Implement the function `closestPointToLineCorrespondences` that computes the corresponding points in \( P \) to those points in \( Q \), using the `point-to-line` matching method.

f) Implement the function `calculateAffineTransformation` that computes the affine transformation matrix needed to align the previously computed corresponding points to the points of \( Q \).

g) Implement the function `applyTransformation` that applies the affine transformation matrix on the points in \( P \).

h) Implement the function `computeError` that computes the error between the points in \( Q \) and the transformed corresponding points.

i) Implement the function `iterateOnce` that performs one iteration of ICP and prints the error of that iteration.

The Gnuplot script `scripts/plot.gp` and the image on the Wiki show \( Q \), \( P \), and the aligned points \( P_1 \) according to the two correspondence strategies.