Link to the repository
https://github.com/Scoop-A-Doop/cs2300-coursework

Pseudocode:
Part1:
- Establish a final variable that will format all numbers to be up to 4 significant digits.
- main: Establish all needed matrices.
- readFile: Goes through the file and only extracts the first 6 numbers and places them accordingly in their respective matrix.
- calculate: This method takes two matrices and goes through the process of calculating Ax=B, solving for x. Steps include flipping A diagonally, calculating its inverse, and multiplying the inverse by B.
- print: This method writes the values of the x matrix (that was calculated in calculate) into a file titled “PA3_P1_Output.txt”.
- Write: Incase “x” that is calculated in calculate is underdetermined or inconsistent, this method will write either “System underdetermined” or “System inconsistent” to the file “PA3_P1_Output.txt”.

Part2:
- Establish a final variable that will format all numbers to be up to 4 significant digits.
- Main: Establish variables and call all functions to calculate the answer.
- calculateLambda: Goes through the entire polynomial factoring process (FOIL) to calculate lambda 1 and lambda 2. If it cannot calculate those two values using FOIL, then use the quadratic equation. If the quadratic equation doesn't work, then there are no real eigenvalues.
- FOIL: This method goes through the FOIL process given an i and j. This method combines two factors (ex: (L+4)(L+2)). This method is used at the end of calculateLambda to make sure the two eigenvalues that it came up with through FOIL are actually valid eigenvalues.
- calculateEigenvector: Calculates the eigenvector. Substitute eigenvalue into the diagonal elements of the matrix, calculate the shear of the matrix and then shear the matrix. Use Gauss Elimination then normalize those values. Save it into a matrix and return the matrix.
- matrixComposition: Calculates the Transpose of R (eigenvectors) then calculate R*Lambda then multiply that sum by R transpose.
- checkIfEqual: Checks to see if the matrix composition is the same as the original matrix A. If it is, return 1, if it is not, return 0.
- readFile: Reads the text file and saves the first two values of the first two rows into a matrix.
- Print: Writes all of the calculation results (Lambda, eigenvector, composition, 1/0 from checkIfEqual) into “PA3_P2_Output.txt”.

Part3:
- Establish a final variable that will format all numbers to be up to 4 significant digits.
- Main: Establish variables and call all functions to calculate the answer.

- calculateArea: Depending on whether the system is 2D or 3D, calculate the area of a triangle given the points in the matrix. Use the appropriate formula/process depending on whether it is 2D or 3D.
- calculateDistance: Take the first two points of the system. Depending on whether it is 2D or 3D, that is your line/plane. The final point is the point we want to find the distance from the line/plane. Depending on whether it is 2D or 3D, use the appropriate formula to calculate the distance.
- readFile: Reads the text file and saves all of the information into the matrix.
- twoDimensionalOrThreeDimensional: Counts how many lines are in the file. If there are 2 lines, then the system is 2D. If there are 3 lines, then the system is in 3D.
- Print: Depending on whether the system is in 2D or 3D, write all of the calculation results into their appropriate file (2D: "PA3_p3_2D_Output.txt", 3D: "PA3_p3_3D_Output.txt")

Resources:
https://mkyong.com/java/how-to-round-double-float-value-to-2-decimal-points-in-java/ (Used for the decimal conversion static final variable used in all three parts)