I) Introduction

The following table summarizes five commonly used mathematical models of nonvertical straight lines.

<table>
<thead>
<tr>
<th>Model (form)</th>
<th>Equation</th>
<th>Given parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Point-slope</td>
<td>( y - y_1 = m(x - x_1) )</td>
<td>Slope ( (m) )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( x-y ) coordinates of a point ( (x_1, y_1) )</td>
</tr>
<tr>
<td>Two-point</td>
<td>( \frac{y - y_1}{x - x_1} = \frac{y_2 - y_1}{x_2 - x_1} )</td>
<td>( x-y ) coordinates of two points ( (x_1, y_1) ) and ( (x_2, y_2) )</td>
</tr>
<tr>
<td>Two-intercept</td>
<td>( \frac{x}{a} + \frac{y}{b} = 1 )</td>
<td>( x )-intercept ( (a) )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( y )-intercept ( (b) )</td>
</tr>
<tr>
<td>Slope-intercept</td>
<td>( y = mx + b )</td>
<td>Slope ( (m) )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( y )-intercept ( (b) )</td>
</tr>
<tr>
<td>General</td>
<td>( Ax + By + C = 0 )</td>
<td>Real constants: ( A, B, ) and ( C )</td>
</tr>
</tbody>
</table>

II) Project Description

Design and implement a program that permits the user to convert from the point-slope, the two-point, or the two-intercept form to either the slope-intercept or the general form. As an example, your program should interact with the user as follows:

Select the convert-from model:
1) Point-slope form (the line's slope and one point are known).
2) Two-point form (two points on the line are known).
3) Two-intercept form (x-intercept and y-intercept are known).
> 2
Enter the \( x-y \) coordinates of the first point separated by a space > 1 1
Enter the \( x-y \) coordinates of the second point separated by a space > 2 5

Select the convert-to model:
1) Slope-intercept form (the line's slope and y-intercept are computed).
2) General form (the three coefficients are computed).
> 2

Convert from the two-point form: \( \frac{y - 1.00}{x - 1.00} = \frac{5.00 - 1.00}{2.00 -1.00} \)
To the general form: \( 4.00x - 1.00y - 3.00 = 0.00 \)
Do another conversion (Y or N) Y
Select the convert-from model:
1) Point-slope form (the line’s slope and one point are known).
2) Two-point form (two points on the line are known).
3) Two-intercept form (x-intercept and y-intercept are known).
> 1
Enter the line’s slope > 4.2
Enter the x-y coordinates of the point separated by a space > 1 1

Select the convert-to model:
1) Slope-intercept form (the line’s slope and y-intercept are computed).
2) General form (the three coefficients are computed).
> 1

Convert from the two-point form: \( y - 1.00 = 4.20(x - 1.00) \)
To the slope-intercept form: \( y = 4.20x - 3.20 \)

Do another conversion (Y or N) Y

Select the convert-from model:
1) Point-slope form (the line’s slope and one point are known).
2) Two-point form (two points on the line are known).
3) Two-intercept form (x-intercept and y-intercept are known).
> 3
Enter the x-intercept > 1.0
Enter the y-intercept > 2.5

Select the convert-to model:
1) Slope-intercept form (the line’s slope and y-intercept are computed).
2) General form (the three coefficients are computed).
> 2

Convert from the two-intercept form: \( \frac{x}{1.00} + \frac{y}{2.50} = 1 \)
To the general form: \( 2.50x + 1.00y = 2.50 \)

Do another conversion (Y or N) N

As a minimum, your implementation should contain the following user-defined functions:

- **get_problem** - displays the user menu, then inputs and returns as the function value the problem number selected.
- **get_2pt** - prompts the user for the x-y coordinates of both points, inputs the four coordinates, and returns them to the calling function through output parameters.
- **get_pt_slope** - prompts the user for the slope and x-y coordinates of the point, inputs the three values, and returns them to the calling function through output parameters.
- **get_2intrcpt** - prompts the user for the x- and y- intercepts, inputs the two values, and returns them to the calling function through output parameters.
- **slope_intrcpt_from_2pt** - takes four input parameters, the x-y coordinates of two points, and returns through output parameters the slope and y-intercept.
intrcpt_from_pt_slope - takes three input parameters, the x-y coordinates of one point and the slope, and returns the y-intercept as the function value.
slope_from_2intrcpt - takes two input parameters, the x- and y-intercepts, and returns the slope as the function value.
general_from_2pt - takes four input parameters, the x-y coordinates of two points, and returns through output parameters the three coefficients: A, B, C.
general_from_pt_slope - takes three input parameters, the x-y coordinates of one point and the slope, and returns through output parameters the three coefficients: A, B, C.
general_from_2intrcpt - takes two input parameters, the x- and y-intercepts, and returns through output parameters the three coefficients: A, B, C.
display_2pt - takes four input parameters, the x-y coordinates of two points, and displays the two-point form line equation with a heading.
display_pt_slope - takes three input parameters, the x-y coordinates of one point and the slope, and displays the point-slope form line equation with a heading.
display_2intrcpt - takes two input parameters, the x- and y-intercepts, and displays the two-intercept form line equation with a heading.
display_slope_intrcpt - takes two input parameters, the slope and the y-intercepts, and displays the slope-intercept form line equation with a heading.
display_general - takes three input parameters, the A, B, C constants, and displays the general form line equation with a heading.

Note: 1) Items to be submitted include: a flowchart and hand examples (paper form), source codes (both electronic and paper form) and executable files (only electronic form).
2) Electronic submission must be done via e-mail to siripong.potisuk@citadel.edu.
3) If possible, all files should be ‘zipped’ together into one big file (using a utility program such as EASYZIP) and attached to your e-mail. Use the following naming convention for your zip file:

    Project_1_your name.zip

4) Your C++ programs must begin with the following three lines of comment:

    /*  Program name: program_name.cpp */
    /*  Description: State the purpose of the program */
    /*  Author: your name */
    /*  Last modified: due_date */