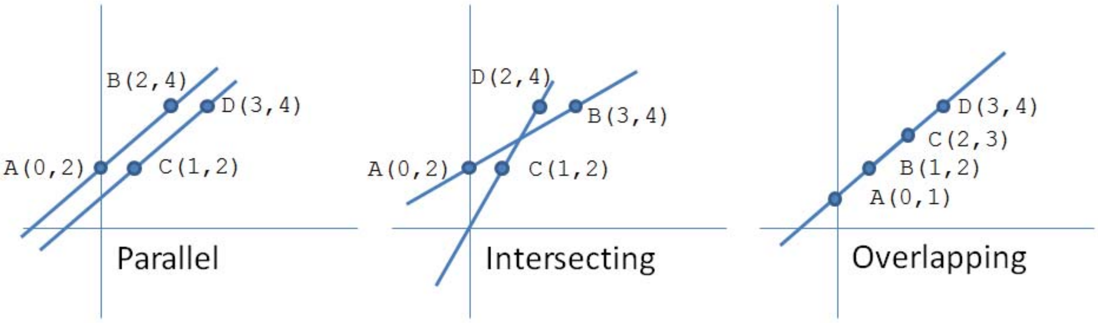
Exercise 1: Lines [50 marks]

Problem Statement

In 2D geometry, given 4 distinct points A, B, C and D on a plane, the two lines, AB and CD, can be parallel (no intersection), intersecting (exactly one intersection) or overlapping (infinite intersections).



One way to determine which case it is (without calculating the number of intersections) is to make use of the **slopes** of AB, CD and AC.

The slope ***k*** of a line can be calculated using the following formula:

$$k = \frac{y_1 - y_2}{x_1 - x_2}$$

where (x_1, y_1) and (x_2, y_2) are the coordinates of two distinct points on the line.

Based on the slopes of the lines, AB and CD are:

- **Parallel** if the slopes of AB and CD are the same but different from that of AC.
- **Intersecting** if the slopes of AB and CD are different.
- **Overlapping** if the slopes of AB, CD and AC are the same.

Write a program to 1) read in the coordinates of the 4 points (all of **int** type), 2) determine whether the two lines AB and CD are parallel, intersecting or overlapping, and 3) print a message accordingly.

Write on the skeleton file **line.c** given to you. You need to include three functions:

- **`determineType()`**: This function takes in the coordinates of 4 points and determines whether the two lines are parallel, intersecting or overlapping.
- **`computeK()`**: This function takes in the coordinates of two distinct points on a line and computes the slope of the line.
- **`printMessage()`**: This function prints a message based on whether the two lines are parallel, intersecting or overlapping.

You are to decide the appropriate parameters and return types for these functions. You may define additional functions as needed.

You need to define constants for the three different types (e.g., 1 for parallel, 2 for intersecting, and 3 for overlapping) and use them in the program.

You may assume that the inputs are valid (i.e., all the coordinates are integers).

You do **not** need to handle the division by zero problem in the computation of slope (i.e., you may assume that x_1 and x_2 are always different).

In addition, due to the relatively large number of inputs for this exercise, you are advised to make use of input redirection to test your program with the given input files.

For example, to run your executable code (e.g., a.out) with an input file (e.g., line1.in), in your UNIX command prompt, enter the following command:

```
a.out < line1.in
```

Sample Runs

Three sample runs are shown below with user input highlighted in **bold**.

```
Enter the coordinates of A: 0 2
Enter the coordinates of B: 2 4
Enter the coordinates of C: 1 2
Enter the coordinates of D: 3 4
The two lines are parallel.
```

```
Enter the coordinates of A: 0 2
Enter the coordinates of B: 3 4
Enter the coordinates of C: 1 2
Enter the coordinates of D: 2 4
The two lines are intersecting.
```

```
Enter the coordinates of A: 0 1
Enter the coordinates of B: 1 2
Enter the coordinates of C: 2 3
Enter the coordinates of D: 3 4
The two lines are overlapping.
```