CS 159 – Homework #2
Due: Monday February 7 at 11:00pm (time local to West Lafayette, IN).
10 Points Possible

Problem: Given as input two points on a first road and a third point on a second road that is perpendicular to the first determine the distance from the points on the first road to that on the second and the coordinate at which the two lines intersect. Both roads form a straight line. All points will be unique and no lines will be vertical or horizontal.

Example Execution #1:

```
Enter coordinates for first point of road -> 0.5 1
Enter coordinates for second point of road -> 5 -2
Enter coordinates for point on perpendicular road -> 4 3

Distance from (0.5, 1.0) to (4.0, 3.0): 4.0
Distance from (5.0, -2.0) to (4.0, 3.0): 5.1
Coordinate of intersection: 2.0, 0.0
```

Example Execution #2 (all numeric data is of the `double` type):

```
Enter coordinates for first point of road -> 4 3
Enter coordinates for second point of road -> -2 -3
Enter coordinates for point on perpendicular road -> 2.5 -2.5

Distance from (4.0, 3.0) to (2.5, -2.5): 5.7
Distance from (-2.0, -3.0) to (2.5, -2.5): 4.5
Coordinate of intersection: 0.5, -0.5
```

Example Execution #3:

```
Enter coordinates for first point of road -> 1 -3
Enter coordinates for second point of road -> 3 0
Enter coordinates for point on perpendicular road -> -1 1

Distance from (1.0, -3.0) to (-1.0, 1.0): 4.5
Distance from (3.0, 0.0) to (-1.0, 1.0): 4.1
Coordinate of intersection: 2.2, -1.2
```

Example Execution #4:

```
Enter coordinates for first point of road -> -5 1
Enter coordinates for second point of road -> -3 -4
Enter coordinates for point on perpendicular road -> -5 0

Distance from (-5.0, 1.0) to (-5.0, 0.0): 1.0
Distance from (-3.0, -4.0) to (-5.0, 0.0): 4.5
Coordinate of intersection: -4.7, 0.1
```

Example Execution #5:

```
Enter coordinates for first point of road -> 5 5
Enter coordinates for second point of road -> 4 0
Enter coordinates for point on perpendicular road -> -6 0

Distance from (5.0, 5.0) to (-6.0, 0.0): 12.1
Distance from (4.0, 0.0) to (-6.0, 0.0): 10.0
Coordinate of intersection: 3.6, -1.9
```

All course programming and documentation standards are in effect for this and each assignment this semester. Please review this document!

Academic Integrity Reminder: Please review the policies of the course as they relate to academic integrity. The assignment you submit should be your own original work. You are to be consulting only course staff regarding your specific algorithm for assistance. Collaboration is not permitted on individual homework assignments.

Additional Requirements:

1. Add the homework assignment header file to the top of your program. A description of your program will need to be included in the assignment header. This particular header can be added to your file by entering `:hhw` while in command mode in `vi`.
2. **Each of the example executions provided for your reference represents a single execution of the program.** Your program must accept input and produce output **exactly** as demonstrated in the example executions, do not add any "bonus" features not demonstrated in the example executions. Your program will be tested with the data seen in the example executions and an unknown number of additional tests making use of meaningful data.
   - All input, output, and the result of any calculation in this program will not exceed what can be stored by a variable of the `double` data type.
   - The use of the `math.h` library is expected.
3. Course standards **prohibit** the use of programming concepts beyond the material found in the first three chapters of the book, notes, and lectures.
4. A program **MUST** compile, be submitted through Vocareum as demonstrated during the lab #0 exercise, and successfully submitted prior to the posted due date to be considered for credit. The C-file you submit must be named exactly: `hw02.c`, no variation is permitted.

Course Programming and Documentation Standards Reminders:

- Indent all code found within the `main` function **exactly** two spaces.
- Place a **single space** between all operators and operands.
- Comment **all** variables to the right of each declaration. Declare only one variable per line.
- Notice that several programs (see program 2-9 on pages 74-75) in the programming text use a single line comment to indicate the start of the local declaration and executable statement sections of the `main` function.
  - At no point during the semester should these two sections ever overlap.
- Select **meaningful identifiers** (names) for all variables in your program.
- Do not single (or double) space the entire program, **use blank lines when appropriate**.
- There is no need to include example output with your submission.

When you submit... only the final successful submission is kept for grading. All other submissions are over-written and cannot be recovered. You may make multiple submissions but only the last attempt is retained and graded.

- Verify in the confirmation e-mail sent to you by the course that you have submitted the correct file to the correct assignment.
- Leave time prior to the due date to seek assistance should you experience difficulties completing or submitting this assignment. All attempts to submit via a method other than through the appropriate assignment on Vocareum will be denied consideration.

Assignment deadlines... are firm and the electronic submission will disable promptly as advertised. We can only grade what you are able submit via Vocareum prior to the assignment deadline.