Problem: Two springs share the same spring constant (newtons per meter) and this is the first input from the user.

- For the first of the two springs you are given from the user (second input) its initial displacement in meters and must calculate its force (newtons) and potential energy (joules).
- For the second spring you are given from the user (third input) its potential energy in joules and must calculate its force (newtons) and its displacement (meters).

Reference for Formulas: [https://en.wikipedia.org/wiki/Hooke%27s_law](https://en.wikipedia.org/wiki/Hooke%27s_law)

<table>
<thead>
<tr>
<th>Force given constant (k) and displacement (x)</th>
<th>Potential Energy given constant (k) and displacement (x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( F = kx )</td>
<td>( PE = \frac{1}{2}kx^2 )</td>
</tr>
</tbody>
</table>

Example Execution #1:
Enter the spring constant for both springs (N/m): 500
Enter the initial displacement of spring #1 (m): 0.55
Enter the potential energy of spring #2 (J): 75.62

Spring #1 Results
Given:
  - Spring constant 500.000000 (N/m)
  - Initial displacement: 0.550000 (m)
Calculated:
  - Force 275.000000 (N)
  - Potential energy 75.625000 (J)

Spring #2 Results
Given:
  - Spring constant 500.000000 (N/m)
  - Potential energy 75.620000 (J)
Calculated:
  - Force 274.990909 (N)
  - Displacement: 0.549982 (m)

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. The example execution provided represents a single test of your program. We will test your final submission with the data used in the test case above and with several unpublished test cases.
   - Your program is expected to accept input and produce output in the exact same manner demonstrated above.
   - All variables should be implemented as the `double` type.
   - You may assume that the user of your program will only enter meaningful values.
   - Do not add any “bonus” features not demonstrated in the example executions provided.

3. Course standards prohibit the use of programming concepts not yet introduced in lecture. For this assignment you can consider all material in the first three chapters of the book, notes, and lectures to be acceptable for use.
   - The use of `sqrt` and `pow` functions found in the `math.h` library are acceptable for use.

4. A program MUST compile, be submitted through the `guru` server as demonstrated during the first week of the semester in lab, and submitted prior to the posted due date to be considered for partial credit.
Course Programming and Documentation Standards Reminders:

- 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. You might consider adopting this habit of commenting the start of each section to help you avoid this mistake.

- Select **meaningful identifiers** (names) for all variables in your program.
- Indent all code found within the `main` function **exactly** two spaces.
- 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 attempt of a 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 (must be named `hw01.c`), to the correct assignment (`hw01`), and to the correct lab section (CRN value).
- 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 guru server as set up during the first lab of the semester will be denied consideration.

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

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.

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