**Problem:** You will create a set of integer values based on two ranges specified as input from the user. Each value in first range will be multiplied by each value in the second range to generate this data set of integers. The user will then specify a target value to locate in the array and your program must report the number of times the target occurs in the array and where the target(s) is (are) found after the data set has been sorted.

First range example: [2, 6]
Second range example: [3, 5]

Data in the set: 6 8 10 9 12 15 12 16 20 15 20 25 18 24 30
Data in the set after sorting: 6 8 9 10 12 12 15 15 16 18 20 20 24 25 30

**Example Execution #1:**

Enter range #1 lower value: 2
Enter range #1 upper value: 6
Enter range #2 lower value: 3
Enter range #2 upper value: 5
Total number of values in data set: 15
Enter target value: 15
Total occurrences found: 2
Locations in the array [6, 7]

**Example Execution #2:**

Enter range #1 lower value: 2
Enter range #1 upper value: 6
Enter range #2 lower value: 3
Enter range #2 upper value: 5
Total number of values in data set: 15
Enter target value: 19
Target not found in the array.

**Example Execution #3:**

Enter range #1 lower value: 2
Enter range #1 upper value: 6
Enter range #2 lower value: 3
Enter range #2 upper value: 5
Total number of values in data set: 15
Enter target value: 30
Total occurrences found: 1
Locations in the array [14, 14]

**Example Execution #4:**

Enter range #1 lower value: 2
Enter range #1 upper value: 6
Enter range #2 lower value: 3
Enter range #2 upper value: 5
Total number of values in data set: 15
Enter target value: 6
Total occurrences found: 1
Locations in the array [0, 0]

---

**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!**
Example Execution #5:

Enter range #1 lower value: 5
Enter range #1 upper value: 2

Error! Upper limit must be >= 5
Enter range #1 upper value: 5
Enter range #2 lower value: -3

Error! Lower limit must be non-negative!!
Enter range #2 lower value: 8
Enter range #2 upper value: 7

Error! Upper limit must be >= 8
Enter range #2 upper value: 10
Total number of values in data set: 3

Enter target value: 40
Total occurrences found: 1
Locations in the array [0, 0]

Example Execution #6:

Enter range #1 lower value: 1
Enter range #1 upper value: 1000
Enter range #2 lower value: 300
Enter range #2 upper value: 300000

Error! Values to be generated exceeds array capacity!

Enter range #1 lower value: 1
Enter range #1 upper value: 1000
Enter range #2 lower value: 14
Enter range #2 upper value: 25
Total number of values in data set: 12000

Enter target value: 13050
Total occurrences found: 3
Locations in the array [8290, 8292]

Additional Requirements:

1. Add the head_hw 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. For this assignment you will be required to implement the user-defined functions (from chapter 4).
3. Failing to follow course standards as they relate to good user-defined function use will result in a zero for this assignment.
   ○ Each user-defined function must represent ONLY a single task in the larger problem.
   ○ A poor use of functions such as combining multiple tasks in a single function will result in a significant loss of points.
4. Revisit course standards as it relates what makes for good use of user-defined functions, what is acceptable to retain in the main function, and when passing parameters by address is appropriate.
5. Your program must accept input and produce output in the exact format as seen in the example executions above.
   - The only input are the four integer values representing the starting and ending values of each range.
   - Example #5 demonstrates the input validation requirement for range starting and ending points.
   - Example #6 demonstrates there is an upper limit of the data set size. This limit is 20,000 integers.

6. Course standards prohibit the use of programming concepts not yet introduced in lecture. For this assignment you can consider all material in the first EIGHT chapters of the book, notes, and lectures to be acceptable for use.
   - Please list your source in the appropriate function header when using or modifying code from an acceptable sources (official text of the course and course lecture notes).
   - All such code must be brought in line with course standards.

7. A program MUST compile to be considered for partial credit. The submission script will reject the submission of any file that does not compile.

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.
- All arrays will be of a declared size that is known before your program is compiled. Use a symbolic/defined constant to represent this value.
- At no point during the semester should the two sections of a function, local declarations and executable statements, be permitted to overlap.
- Select meaningful identifiers (names) for all variables in your program.
  - It is acceptable to use simple identifiers for loop control variables used to traverse the index values of an array.
- All code found inside of relevant selection and repetition constructs must be indented two additional spaces.
- The use of { and } is required with all selection and repetition constructs that have a body.
- Use the course function header (head_fx vi shortcut hfx while in command mode) for every user-defined function in your program.
  - List and comment all parameters to a function, one per line, in the course function header.
  - All function declarations will appear in the global declaration section of your program.
  - The user-defined function definitions will appear in your program after the main function.

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, to the correct assignment (hw07), and to the correct lab section.
- 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 week 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.