CS 159 – Homework #5
Due: Monday March 28 at 11:00pm (time local to West Lafayette, IN).
10 Points Possible

Problem: For an integer `x`, a Boiler Number is the sum of all integers from 1 to `x` inclusive. Given a starting integer `y` and ending integer `z`, display either the even or odd Boiler Numbers generated from the values in the range from `y` to `z` inclusive.

Example Execution #1:

```
Enter 0 for even or 1 for odd values -> 0
Enter starting value -> 1
Enter ending value -> 8

Even Boiler numbers: 6 10 28 36
```

Example Execution #2:

```
Enter 0 for even or 1 for odd values -> 1
Enter starting value -> 1
Enter ending value -> 8

Odd Boiler numbers: 1 3 15 21
```

Example Execution #3 (input validation for first input):

```
Enter 0 for even or 1 for odd values -> 2

Error! Select a valid option!!

Enter 0 for even or 1 for odd values -> -1

Error! Select a valid option!!

Enter 0 for even or 1 for odd values -> 0
Enter starting value -> 20
Enter ending value -> 30

Even Boiler numbers: 210 276 300 378 406
```

Example Execution #4 (input validation for third input):

```
Enter 0 for even or 1 for odd values -> 1
Enter starting value -> 100
Enter ending value -> 100

Error! Ending value must be > 100

Enter ending value -> 115

Odd Boiler numbers: 5151 5253 5565 5671 5995 6105 6441 6555
```

Example Execution #5 (input validation for second input):

```
Enter 0 for even or 1 for odd values -> 0
Enter starting value -> 0

Error! Starting value must be positive!

Enter starting value -> 340
Enter ending value -> 341

Even Boiler numbers: 57970
```

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.

Example Execution #6:

```
Enter 0 for even or 1 for odd values -> 0
Enter starting value -> 1
Enter ending value -> 2

Error: No values generated.
```

Example Execution #7:

```
Enter 0 for even or 1 for odd values -> 1
Enter starting value -> 7
Enter ending value -> 8

Error: No values generated.
```

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.
   - All input given to test your program and all values generated during its execution will not exceed that which can be stored in an `int` variable.
   - Expectations for input validation are demonstrated in the example executions.

3. For this assignment you will be **required** to implement the user-defined functions (from chapter 4). Failing to follow course standards as they relate to good user-defined function use will result in a **zero for this assignment.**

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.**
   - In many cases user-defined function use should result in a `main` function that only declares variables and makes function calls.

5. Course standards **prohibit** the use of programming concepts not yet introduced in lecture. For this assignment you can consider all material in the **first SIX chapters** of the book, notes, and lectures to be acceptable for use.
   - Course standards **prohibit** the use of programming concepts beyond the material found in the first SIX chapters of the book, notes, and lectures. No arrays!

6. 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: `hw05.c`, no variation is permitted.

Course Programming and Documentation Standards Reminders:

- Code found inside the body of relevant selection and repetition constructs must be indented two additional spaces.
- Make use of `{` and `}` with all relevant selection and repetition constructs.
- See page 258 of your C programming text regarding the proper indentation for a `switch` construct.

- Use the course function header (`vi` shortcut `:hfx` while in command mode) for every user-defined function..
  - 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.**
- Indent all code found within the `main` and all user-defined functions **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 a function.
  - At no point during the semester should these two sections ever overlap.

- Select **meaningful identifiers** (names) for all variables in your program.

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.