

CS 15900 – Homework 5

Due Monday 25 March 2024 at 11:00 PM (time local to West Lafayette, IN).

10 Points

Problem: Given a seed for the random number generator and limits for the range from which the number will be selected (inclusive of the end points), create the number and manipulate it to produce additional values by moving the digit in the ones place to the highest power of ten and counting the number of times a value larger than the original is identified.

- For example: The original value is 42785, from this the following are created: 54278, 85427, 78542, 27854. Three of these values are larger than the original.

Example Execution #1 (explained above):

```
Enter the desired seed value -> 2000
Enter the desired minimum value -> 30000
Enter the desired maximum value -> 90000
```

```
Larger numbers generated from 42785: 54278 85427 78542
Count of larger numbers generated from 42785: 3
```

Example Execution #2:

```
Enter the desired seed value -> 3000
Enter the desired minimum value -> 123456
Enter the desired maximum value -> 987654
```

```
Larger numbers generated from 422428: 842242 428422
Count of larger numbers generated from 422428: 2
```

Example Execution #3:

```
Enter the desired seed value -> 4000
Enter the desired minimum value -> 1000000
Enter the desired maximum value -> 5500000
```

```
Larger numbers generated from 2612495: 5261249 9526124 4952612 6124952
Count of larger numbers generated from 2612495: 4
```

Example Execution #4:

```
Enter the desired seed value -> 5000
Enter the desired minimum value -> 3
Enter the desired maximum value -> 100
```

```
Larger numbers generated from 29: 92
Count of larger numbers generated from 29: 1
```

Example Execution #5:

```
Enter the desired seed value -> 6000
Enter the desired minimum value -> 90
Enter the desired maximum value -> 1400
```

```
Larger numbers generated from 369: 936 693
Count of larger numbers generated from 369: 2
```

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 seek assistance from course staff members ONLY. Collaboration is not permitted on individual homework assignments.

Example Execution #6 (how to handle when no larger numbers are created):

```
Enter the desired seed value -> 7000
Enter the desired minimum value -> 0
Enter the desired maximum value -> 9
```

```
Count of larger numbers generated from 2: 0
```

Example Execution #7 (minimum and maximum value validation requirements demonstrated):

```
Enter the desired seed value -> 500
Enter the desired minimum value -> -10
```

```
Error! Minimum must be a non-negative integer.
```

```
Enter the desired minimum value -> 25000
Enter the desired maximum value -> 20000
```

```
Error! Maximum value must be greater than 25000.
```

```
Enter the desired maximum value -> 30000
```

```
Larger numbers generated from 28670: 70286 67028 86702
Count of larger numbers generated from 28670: 3
```

Example Execution #8 (largest value in range generated):

```
Enter the desired seed value -> 1305
Enter the desired minimum value -> 25000
Enter the desired maximum value -> 27139
```

```
Larger numbers generated from 27139: 92713 39271 71392
Count of larger numbers generated from 27139: 3
```

Example Execution #9 (smallest value in range generated):

```
Enter the desired seed value -> 5478
Enter the desired minimum value -> 42685
Enter the desired maximum value -> 50000
```

```
Larger numbers generated from 42685: 54268 85426 68542
Count of larger numbers generated from 42685: 3
```

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 will be integer values and all values generated during the described process will remain within the limits of the `int` data type.
 - Input validation expectations for minimum and maximum values are demonstrated in the seventh example execution. **All values must be non-negative, including the seed, the desired maximum value must be greater than the desired minimum value.**
3. For this assignment you will be **required** to implement 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**.

Additional Requirements (continued):

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 beyond the material found in the first **SIX** chapters of the book, notes, and lectures. The use of any arrays would violate the requirements of this assignment and would result in no credit being awarded for your effort.
6. A program **MUST** compile, be submitted through Vocareum as demonstrated during the Lab #0 exercise, and submitted prior to the posted due date to be considered for credit. The source code 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 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.
- Maximize your use of symbolic/defined constants and minimize your use of literal constants.
- 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.

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

- Verify in the confirmation email 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.