**Homework #7**

**Due Wednesday, April 18**

Write a program that tests integers to see if they are *rising* numbers. A *rising* number is one in which all digits are in nondecreasing order. By this, I mean that the leftmost digit may be anything and all other digits must be equal to or larger than the digit immediately to their left. For example, 25569 is a rising number and 314 is not a rising number (1 is not equal to or larger than 3). The program should allow testing as many values as the user wishes. Also, for each input value (whether rising or not), the program should output the value of the next-larger rising number (the smallest rising number that is greater than the user’s input).

Your program should be designed using functions for key (hierarchical) ideas. Use my tool functions as appropriate. In addition to functions for the main control structure of your program (inputting doing again, etc.), the program should include and use the following two functions, which are program-specific “math”-like functions – they take a parameter and produce a single value. They do *no* console input/output:

* One function should receive an integer parameter and return a Boolean value that indicates whether or not the parameter is a rising number. Note: x%10 gives the value of the rightmost digit of x. x/10 gives all digits *except* the rightmost digit (thus, x = x/10) *removes* one digit from x). Using these in a loop allows you to look at each in a number. By remembering the *previous* digit each time around, you can check for the desired property.
* The other function should receive an integer parameter and return an integer which is the *next* rising number after the parameter. To find the next rising number, use a loop that counts up from the parameter’s value, testing each value (using the rising number test function described above), until a rising number is found. Do *not* “construct” the next rising number by gluing digits together.

The executable program **Gold07.exe** demonstrates what your program should do.

You should turn in (in a pocket folder): this assignment/grading sheet (write your name in the space provided), your statement of completeness, a structure chart with *function* names in the boxes, and a full printout of your program. All documents should be appropriately labeled. Also, place a copy of your main.cpp file in a properly named folder, into your private FTP folder.

**Grading Sheet, Homework #7**

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Criteria Possible Achieved**

Statement of Completeness 5 \_\_\_\_\_\_\_\_

Structure Chart 5 \_\_\_\_\_\_\_\_

Clear Indentation and Spacing 5 \_\_\_\_\_\_\_\_

Internal Code Comments 5 \_\_\_\_\_\_\_\_

Clear Identifiers 5 \_\_\_\_\_\_\_\_

Appropriate Use of Statements & Expressions 5 \_\_\_\_\_\_\_\_

Appropriate Use of Functions 10 \_\_\_\_\_\_\_\_

Complete/No Errors 5 \_\_\_\_\_\_\_\_

Output Format and Correctness 5 \_\_\_\_\_\_\_\_

Presentation 3 \_\_\_\_\_\_\_\_

Total: 53 \_\_\_\_\_\_\_\_

Notes: