C Programming The Basics

For this assignment, you will provide the missing piece to complete the Caesar Cipher example from the course notes. In particular, you will implement the following C function:

```
//////////////////////////////////////////////////////////////// applyShift()
// applyShift() computes the appropriate shift for an alphabetic character.
//
// Note: non-alphabetic characters are not shifted, so applyShift() will
//       simply return the original character unless it is alphabetic.
//
// Restrictions:
//    You may not alter the given interface for the function.  Doing so will
//    almost certainly result in compile-time errors with the testing code.
//
//    Full credit will not be given for any solutions that uses a loop
//    construct, or an array within the body of applyShift() or any helper
//    function you may write.
//
//    The exact penalty for these will be determined after the assignment is
//    closed
//
//    Aside from the explicit restrictions above, you may use any C language
//    features and any C Standard Library functions you like.
//
// Parameters:
//    Original   the character that needs to be shifted
//    shiftAmt   the number of positions Original must be shifted
//
// Pre:
//    Original is initialized to the appropriate character (ASCII encoding)
//    shiftAmt is initialized to an integer value (no restrictions)
//
// Returns:  the character (ASCII encoding) that would result from shifting
//           Original by shiftAmt positions
//
char applyShift(char Original, int shiftAmt) {

   . . .
}
```

Download the posted shell file, `applyShift.c`, and implement your solution within that file. This will require simple separate compilation, which is explained in the comments for the posted example code.

You should download the posted example code for the Caesar Cipher program and make sure that your solution compiles with the posted code. Follow the instructions in the comments for the example code to build an executable.

You may need to add `include` directives to the file as needed for any C Standard Library features you use. You may write secondary "helper" functions if you like; if so, those must be defined and declared within the file `applyShift.c`.

Take note of and conform to the restrictions in the header comment above.

Testing your solution is your responsibility; try to think of as many logically-distinct cases as possible and be sure your solution handles all of them. Refer to the course notes on the Caesar Cipher example to see how to run your program with an input file; it will be useful to create different input files for the different test cases you identify.

What to Submit

You will submit your `applyShift.c` source file (not compressed).

This assignment will be graded automatically. You will be allowed up to ten submissions for each part of this assignment, so use them wisely. Test your programs thoroughly before submitting them. Make sure that your programs produce correct results for every logically valid test case you can think of. Do not waste submissions on untested code, or on code that does not compile with the Caesar Cipher example code from the course website.

The Curator will assign a score based on runtime testing of your submission; your best score will be counted; the TAs will later verify that your best submission meets the stated restrictions, and assess penalties if not.

The *Student Guide* and other pertinent information, such as the link to the proper submit page, can be found at:

http://www.cs.vt.edu/curator/

Pledge:

Each of your program submissions must be pledged to conform to the Honor Code requirements for this course. Specifically, you **must** include the following pledge statement in the submitted file:

```
##    On my honor:
##
##    - I have not discussed the C language code in my program with
##      anyone other than my instructor or the teaching assistants
##      assigned to this course.
##
##    - I have not used C language code obtained from another student,
##      or any other unauthorized source, either modified or unmodified.
##
##    - If any C language code or documentation used in my program
##      was obtained from another source, such as a text book or course
##      notes, that has been clearly noted with a proper citation in
##      the comments of my program.
##
##    <Student Name>
```

Failure to include this pledge in a submission is a violation of the Honor Code.