CSCI 689 **Computer Assignment 3** **Spring 2016**
(30 points)

Caesar Cipher

The encryption technique called **Caesar cipher** replaces each letter in a text message by a letter that appears a fixed distance in the alphabet. As an example, suppose that you wanted to encode a message by shifting every letter ahead three places. In this cipher, each A becomes a D, B becomes E, and so on. If you reach the end of the alphabet, the process cycles around to the beginning, so that X becomes A, Y becomes B, and Z becomes C.

Note that the transformation applies only to letters; any other characters are copied unchanged to the output. Moreover, the case of letters is unaffected; lowercase letters come out as lowercase, and uppercase letters come out as uppercase. You should also write your C++ program so that a negative value of shift means that letters are shifted toward the beginning of the alphabet instead of toward the end.

There are two data files for this program: prog3.d1 and prog3.d2. Both are located in directory: /home/cs689/progs/16s/p3. The 1st file contains several test values for shift and the 2nd one contains a text message to encode.

In addition to your C++ source file, you also need to have a header file, where you put declarations of all constants (number of letters in the alphabet for the text message and the full path name of the 2nd data file) and function prototypes that you use in your program.

The main ( ) routine basically calls the following function for each shift value entered from the stdin.

- void process_infile (int shift): This function first opens the 2nd data file, and if it is unsuccessful, it calls the error ( ) function to display the error message. It prints out the shift value on stdout passed as an argument and gets the text input from the data file. To process each input line in the data file, it calls the encodeCaesarCipher ( ) function, which is described below, and it prints out the encrypted text returned by this function on stdout. Finally, it closes the data file.

To implement a Caesar cipher, you should define the following function.

- string encodeCaesarCipher (string str, int shift): It returns a new string formed by shifting every letter in str forward the number of letters indicated by shift, cycling back to the beginning of the alphabet if necessary. To implement shifting, this function calls the following auxiliary function.

- int new_position (char c, int shift): For a given character c and a positive or negative shift value, it returns the new position of c after shifting.

- void error (const string& msg): This routine prints the error message msg on stderr and exits the program with the nonzero return value EXIT_FAILURE.

At the top of your source file, include its header file. To compile your source file and link the created object file with the system library routines, create a makefile. Insert the statements in this file as follows. Then, execute the make command as: make T=prog3 (assuming you name your source and header files as prog3.cc and prog3.h).

```
OPT = -std=c++11 -c -g -Wall -Wextra          # compiler options
D1 = /home/cs689/progs/16s/p3/$T.d1           # several shift values

$T.exe: $T.o
        g++ -o $@ $T.o

$T.o: $T.cc $T.h
        g++ ${OPT} $T.cc

execute:
        $T.exe < ${D1} 2>&1 | tee $T.out

clean:
        /bin/rm -f *.o *.exe
```

For a final test of your program, execute: make execute T=prog3. When the execution is over, the output file prog3.out will contain the encrypted text for each shift value in file prog3.d1, as well as any error messages might be generated by your program. See the correct output in file prog3.out in directory: ~cs689/progs/16s/p3. After you are done, delete the object and executable files by executing: make clean.

Submit your source and header files of your program to your TA, executing: mail_prog prog3.cc prog3.h. When the shell script mail_prog prompts, e*nter 2 for the section number and 3 for the program number.*