

# ECE 71 – Engineering Computations in C

Professor Kriehn – Fall 2017

**Code Due By:** Midnight on Tuesday, November 7, 2017

**Writeup Due By:** Class on Wed/Thu, Nov 8/9, 2017

## HOMEWORK #28 – Phone Number Translator

Write a program that translates an alphabetic phone number into numeric form using C-based character arrays:

```
Enter Phone Number:    CALLATT
Numeric Form:          2255288
```

In case you do not have your cell phone handy, remember that: 1 = .,@, 2 = ABC, 3 = DEF, 4 = GHI, 5 = JKL, 6 = MNO, 7 = PQRS, 8 = TUV, 9 = WXYZ. If the original phone number contains non-alphabetic characters besides the ones listed for the number 1 (‘.’, ‘,’, or ‘@’), your program should leave the results unchanged. For example:

```
Enter Phone Number:    1-800-COL-LECT
Numeric Form:          1-800-265-5328
```

You may assume that any letters entered into your program by the user could upper or lower case.

### Specifications:

The program needs to store the original phone number in array of characters. The program also needs to store the numeric number in a second array of characters. You may assume that the phone number is no more than 15 characters long, and any letters that are entered into the program by the user may be upper or lower case. Define a constant N for your array size and use it throughout your program.

If you execute the program, the following information should be displayed:

```
~> main.o
Enter Phone Number:    1-800-765-5234
Numeric Form:          1-800-765-5328
~> main.o
Enter Phone Number:    1800KrIeHn7
Numeric Form:          18005743467
~> main.o
Enter Phone Number:    011886INTERNTL.
Numeric Form:          011886468376851
~>
```

**HINT:** You will need to initialize your two arrays with the NULL (‘\0’) character (ASCII Code 0) so that if you do not print out 15 numbers, you will not print out junk values left over in other elements in the arrays. You will need to use a `while()` loop to read in your characters, and use a `for()` loop for your conversion. You may also find a `switch()` statement to be extremely useful...

## HOMEWORK #29 – Sentence Character and Word Counter

Write a program that will read in a line of text (use the `getline()` function) into a string variable and output the number of words in the line and the number of occurrences of each letter.

### Specifications:

Define a word to be any string of letters that is delimited at each end by either whitespace (not just the space character!), a period, a comma, or the beginning or end of the line. You can assume that the input consists entirely of letters, whitespace, commas, and all forms of punctuation. When outputting the number of letters that occur in a line, be sure to count uppercase and lowercase versions of a letter as the same letter. Output the letters in alphabetical order in uppercase and list only those letters that occur in the input line.

If you execute the program, the following information should be displayed:

```
~> main.o
Enter a Line of Text: I say, "Hi!".

Words:      3

Letters:
A    1
H    1
I    2
S    1
Y    1
~>
```