APS106: FUNDAMENTALS OF COMPUTER PROGRAMMING
LAB # 8 - MONDAY, MARCH 17, 2:00 – 4:00

This lab will test your ability to declare arrays, initialize arrays, and compute arrays using function calls. The extension of your source file must be .c, not .cpp (e.g. lab8.c).

Background: In May of 1844, Samuel F. B. Morse sent the message "What hath God wrought!" by telegraph from Washington to Baltimore, heralding the beginning of the age of electronic communication. To make it possible to communicate information using only the presence or absence of a single tone, Morse designed a coding system in which letters and other symbols are represented as coded sequences of short and long tones, traditionally called *dots* and *dashes*. In Morse code, the 26 letters of the alphabet are represented by the following codes:

A	.-	J	.---	S	...
B	-...	K	-.-	T	-
C	-.-.	L	.-..	U	..-
D	-..	M	--	V	...-
E	.	N	-.	W	.--
F	..-.	O	---	X	-..-
G	--.	P	.--.	Y	-.--
H	....	Q	--.-	Z	--..
I	..	R	.-.		

You can easily store these codes in a program by declaring an array of 26 by 5 characters (`char morse[26][5]`) and storing the sequence of characters corresponding to each letter in the appropriate array entry (from `'A'` to `'Z'`).

Problem: Write a program that reads a string from the user and translate each letter in the string to its equivalent in Morse code, using periods to represent *dots* (.) and hyphens (-) to represent *dashes*. Separate words in the output by **replacing each space** in the input with a newline character. Assume that the user will enter a string **without** punctuation. Your program must have the following two functions:

```
void getText(char a[]);           /* get a string from user */

void printMorseCode(char a[], char b[][5]); /* print Morse equivalent
                                               of array a[] using code
                                               stored in array b[][5] */
```

Write a driver (main function) to test your program. The arrays `a[80]` and `morse[26][5]` should be declared in the `main()` function and passed as arguments. **YOU MUST INCLUDE** <string.h>

See hints on next page…

Hints:

Declare array `a[]` as `char a[80]` and use `scanf("%[^!]", a)` to read the string from the keyboard. The string should NOT contain any punctuation and MUST be terminated with an exclamation mark `(e.g.: What hath God wrought!)`. `scanf("%[^!]", a)` will read the string up to the exclamation mark (exclamation mark excluded) and will put the actual string in the array `a[]`. You can then get access to each character of the string using a `for` loop:

```
for(i = 0; i < strlen(a); i++) /* i is the position in a[i] */
```

Make sure the `getText()` function converts the string obtained from the user to ALL UPPER CASE.

Initialize the array `morse[26][5]` in the `main()` function the following way:

```
                    /* A         B          C         D ...     */
char morse[26][5] = {{".-"}, {"-..."}, {"-.-."}, {"-.."}, etc... }
```

To print the string at position `i` in the array `morse[26][5]`,
use `printf("%s ", morse[i])`. Do not specify the other dimension.