CS1342 – Spring 2023

Program 2

In this program, you will utilize C-Strings and Functions to create a program that is able to convert text into Morse Code! Your program must be submitted to GitHub by 6:00 AM CST on **Monday, February 20, 2023.** You will receive a 30 point penalty for any program submitted within 48 hours after the due date. Any programs submitted after 6:00 AM CST on **Wednesday, February 22, 2023** will receive a 0.

Overview

Morse code is a popular communication method primarily used in telecommunications to encode text characters as standardized sequences of 2 different signals, called *dots* or *dashes.* Morse code was inventory by Samuel Morse, who invented the telegraph in which Morse code was used.

Today it is still a common form of communication as it requires very little technology and can used and interpreted through different types of sense, e.g. sight, sound, and touch.

In your program, you will create 2 main functions:

1. Convert text to morse code
2. Convert morse code to text

Program Specifications

You are to create a program that begins by prompting the user to enter in a *command* to signify whether they would like to convert text to morse code, convert morse code to text, or exit.

```
Enter in a command
>
```

The three different possible commands are MORSE, ALPHA, and EXIT. If a user should enter in any other value, please tell the user they have entered an incorrect command and then re-prompt the user.

MORSE command

When a user enters the command MORSE, your program should prompt the user to enter up to 250 characters of text, ending in a period (.). Your program will then convert that text into morse code and print it to the screen.

The only possible characters a user will enter will be any of the following set of character:

A – Z, a – z, . (period)

Where capital letters and lower case letters map to the same morse code value. In this program the “.” Character will be represented by the symbol “!”.

A full morse code conversion table for A – Z can be found here: https://www.electronics-notes.com/articles/ham_radio/morse_code/characters-table-chart.php

```
Enter in a command
> MORSE
Enter in alpha characters to be converted to morse code:

the quick brown fox jumped over the lazy dog. [ENTER]

- .... . / --.- ..- .. -.-. -.- / -... .-. --- .-- -. / ..-. ---
-..- / .--- ..- -- .--. . -.. / --- ...- . .-. / - .... . / .-..
.- --.. -.-- / -.. --- --. !
```

ALPHA command

When a user enters the command ALPHA, your program should prompt the user to enter up to 1000 characters of morse code text, which will always end in a “!” (representing a period). Your program will then convert the morse code text into alpha characters.

```
Enter in a command
> ALPHA
Enter in morse code to be converted to alpha characters

- .... . / --.- ..- .. -.-. -.- / -... .-. --- .-- -. / ..-. ---
-..- / .--- ..- -- .--. . -.. / --- ...- . .-. / - .... . / .-..
.- --.. -.-- / -.. --- --. !

the quick brown fox jumped over the lazy dog.
```

EXIT command

When a user enters the EXIT command, your program should be terminated .

Implementation Requirements

Make sure to abide by the following requirements for full credit:

- Your program should have at least 3 functions
    - 1 function should use **pass by value**
    - 1 function should use **pass by reference** (using a c-string as a parameter counts!)
    - 1 function should return a value (be non-void).
- The ONLY libraries you are allowed to use from the STL are **<iostream>** and **<cstring>**
    - **15 points** will be taken off per use of any other library.
- Your function prototypes and definitions should be properly written in a header and source file (.h / .cpp). No function definitions should appear in main.cpp

Hints

You may find the usage of the following functions useful for this program:

- `cin.getline(arg1, arg2)` – Used to read in *arg2* number of characters and store into a c-string *arg1.*
    - *Ex:*

```
const int size = 10;
char my_cstring[size]; // declare a c-string of max size 10

// Read in max of 10 characters from terminal into my_cstring
std::cin.getline(my_string, size);
```

- `strlen(cstring)` – used to find the number of characters in a cstring
- `strcmp(cstring1, cstring2)` – used to compare the values of 2 different cstrings
- `strcpy(cstring1, cstring2)` – used to copy the contents of one cstring variable or literal into another.

BONUS (+15 points)

An extra **15 points** will be awarded to students who complete this program without the use of the **<cstring>** library. NOTE: This will require you to implement your own versions of the *strlen, strcmp, and strcpy* functions. Before attempting the bonus, ensure you have a completed program with the use of the <cstring> library functions.