

EXERCISE 3

There are three parts to this exercise. Follow the instructions for each part below and turn in your submission according to the instructions for each exercise.

EXERCISE 3.1

Programming exercise 3.1 requires you to write in assembly language the equivalent function to the C Standard Library function `strlen()`. The `strlen()` function returns the length of a string passed to it in a single argument:

```
int strlen(char *string);
```

Furthermore, the function must be callable from a C program. For example, the following lines of code in C should produce the same output:

```
printf("The string is %d characters long.\n", strlen(testString));  
printf("The string is %d characters long.\n", mystrlen(testString));
```

Note that the second `printf()` function is calling `mystrlen()` which is your assembly language version of `strlen()` that you must write.

You will find a starter project titled `Exercise_3_1` on the Week 5 D2L page. This project provides what you need to compile and link your program. When finished, please turn in your version of `mystrlen.asm` to the D2L site for Coding Exercise #3.

EXERCISE 3.2

In this exercise you will write an assembly language function to find the largest string in a list of strings. The function has two parameters. The first is a pointer to a list of strings. The second is a count of the number of strings in the list. In C, the prototype of the function is as follows:

```
char *FindLongestString(char *pStringList[], int stringCount);
```

You can assume the string list parameter is an array of pointers to the strings that make up the list. For example, the following list illustrates the structure of the list:

```
char *stringList[3];  
stringList[0] = "You can't see the forest through the trees";  
stringList[1] = "Try not. Do or do not. There is no try";
```

```
stringList[2] = "Shall we play a game?";
```

The function must return a pointer to the largest string in the list. Using the above example, the function should return a pointer to the first string in the list.

Begin the assignment with the provided Exercise_3_2 project. This project contains a C program that sets up the string list and makes the call to the FindLongestString function. There is also a C implementation of the function that demonstrates the expected output. Note that you may use either strlen() or the mystrlen() function that you wrote in Exercise 3-1 to complete this exercise.

When finished, please turn in your version of FindLongestString.asm to the D2L site for Coding Exercise #3.

EXERCISE 3.3

In this exercise you will write an assembly language function to print the contents of a structure. The function is called PrintRecord and its sole parameter is a pointer to the structure containing the data to print. The structure is defined as follows:

```
#pragma pack(1)
typedef struct student_record
{
    char firstName[64];
    char middleName[64];
    char lastName[64];
    int class;
    int studentId;
} StudentRecord;
```

The function must print the structure's contents as follows:

```
First Name: Susan
Last Name: Anthony
Class:    5
Student Id: 12345
```

Being the assignment with the provided Exercise_3_3 project. This project contains a C program that defines the structure, populates the structure, and makes the call to the PrintRecord function. There is also a C implementation of the function that demonstrates the expected output.

When finished, please turn in your version of PrintRecord.asm to the D2L site for Coding Exercise #3.
