Lab 12: More string manipulation

Goals

Build C functions capable of performing additional operations upon null-terminated text strings.

Background

Strings

- Character strings (or simply strings) are arrays of char variables.
- They are used in a C language program to store words or sentences in the English (or some other) language.
- Some programming languages have an actual variable type called string, but C does not.
- The logical end of a C language string is denoted by the presence of a byte consisting of all 0 bits some times called a NULL byte.
- The program explicitly must provide storage for the NULL byte.
- Thus, a character array that can hold the string "Hello" must be at least 6 bytes in size: char hello[6];

Previously assigned string functions

In lab 6 you constructed the following string functions:

- int my_strlen(const char str[]);
- int my_strcpy(const char str1[], char str2[]);

Note that my_strcpy() is not compatible with the real strcpy:

- char *strcpy(char *dest, const char *src);

Assignment:

In this lab we will create two more string functions that will be fully compatible with their "official" counterparts. We will also use pointer notation. Use of array notation will lead to a deduction. Write a program called lab12.c that contains two functions as follows:

1. Create a function `char *my_strcat(char *dest, const char *src);` This function will catenate the string pointed to by *src to the end of the string pointed to by *dest. The function must return the original value of dest. Two while() loops should be used. The first should find the end of the string pointed to by dest and the second should be used to catenate the string pointed to by src. Be sure that the new destination string is properly terminated with a NULL byte.

2. Create a function `int my_strcmp(const char *s1, const char *s2);` that will compare two strings for equality. You will need one while() loop here and it must perform character at a time comparison. If the two strings are equal (have the same length and each pair of bytes has the same value) it should return 0. Otherwise -1 should be returned if, at the first mismatch, the value in s1 is less than the value in s2, and 1 should be returned in the value in s1 is greater than the value in s2.

Here is a starting point for a main() function that could be used for testing. DO NOT TURN IN THE MAIN FUNCTION.

```
     #include <stdio.h>
     char *my_strcat(char *dest, const char *src);
     int my_strcmp(const char *s1, const char *s2);
     int main()
     {
        char t0[20] = {0};
        char *t1 = "abc";
        char *t2 = "abcd";
        char t3[20] = "efg";
        int v1;
        int v2;
        int v3;
        my_strcat(t0, t1);
        my_strcat(t3, t0);
        v1 = my_strcmp(t1, t2);
        v2 = my_strcmp(t3, t1);
        v3 = my_strcmp(t2, t2);
     }
```

Turn In Work

Show your TA that you completed the assignment. Then turn in your lab12.c program using the command:

```
sendlab.101.section_number 12 lab12.c
```