

## Problem Set 4 Exercise #07: My String Functions

**Reference:** Lecture 10 notes

**Learning objective:** Characters and strings

**Estimated completion time:** 60 minutes

### Problem statement:

In the lecture, we have introduced several string functions such as: **strlen**, **strcpy**, **strcat** and **strcmp**. Now we are to implement our own version of string functions of the same functionalities. Of course, you are not allowed to use any string functions from `<string.h>` when writing your own version. However, your functions may invoke each other if necessary.

Write a program **my\_str\_functions.c** that contains the following functions.

- (a) **int my\_strlen(char s[])** that returns the number of characters in string **s** before (and excluding) the termination null character.
- (b) **void my\_strcpy(char s1[], char s2[])** that copies string **s2** to **s1**, including the terminating null character, stopping after the null character has been copied.
- (c) **void my\_strcat(char s1[], char s2[])** that appends a copy of string **s2**, including the terminating null character, to the end of **string s1**. The first character of **s2** overrides the first null character of **s1**.
- (d) **int my\_strcmp(char s1[], char s2[])** that compares two strings character by character, according to the ASCII character ordering. The function returns 15, 0, or -15, if string **s1** is greater than, equal to, or less than string **s2**. (C language specifies return values to be positive or negative, if the first string is greater or less than the second string. Here we use 15 and -15 as example.)

You may assume that both **s1** and **s2** contain less than 100 characters.

### Sample run #1:

```
Enter s1: abc
Enter s2: abcd
my_strlen(s1) returns 3
my_strlen(s2) returns 4
my_strcmp(s1, s2) returns -15
my_strcat(s1, s2) returns abcabcd
my_strcpy(s1, s2) returns abcd
```

### Sample run #2:

```
Enter s1: ac bd
Enter s2: ab cd
my_strlen(s1) returns 5
my_strlen(s2) returns 5
my_strcmp(s1, s2) returns 15
my_strcat(s1, s2) returns ac bdab cd
my_strcpy(s1, s2) returns ab cd
```