

# SysProg Assignment 1: String Functions

Arno Bakker

Guillaume Pierre

**Deadline: 12 September 2016 at 13:29**

---

This assignment *must* be submitted via Blackboard.

Assignments *must* contain:

- Two source files called `mystrlen.c` and `mystrcmp.c`
- A text file called `README` (no extension, all in capitals) containing your full name, student number and email address.


Assignments *must* be packaged into an `uncompressed tar file` with the files in the top-level directory (no subdirectories!). It is not necessary to submit a Makefile.

If your submission does not adhere to these requirements it will be refused!

---

## 1 String length

`strlen()` is a standard function from the C library that computes the length of a character string. The goal of this exercise is to program this function yourself. Type the code skeleton below and call this file `mystrlen.c`. Then, fill in function `mystrlen()`. Of course, it is forbidden to use function `strlen()` or another similar one in your own implementation of `mystrlen()`.

```
#include <stdio.h>

int mystrlen(char *input_string) {
    /* This function returns the length of the input string */

    /* WRITE FUNCTION CODE HERE! */
}

int main(int argc, char **argv) {
    int length;
    if (argc!=2) {
        printf("Usage: strlen <input_string_with_no_space_inside_it>\n\n");
        return 1;
    }
    length = mystrlen(argv[1]);
    printf("The length is: %d characters.\n",length);
    return 0;
}
```

An example execution looks as follows:

```
$ ./strlen
Usage: strlen <input_string_with_no_space_inside_it>

$ ./strlen foo
The length is: 3 characters.
$ ./strlen foobarbaz
The length is: 9 characters.
$ ./strlen ""
The length is: 0 characters.
$
```

To enable automatic testing of your submission, your program *must* use the output format "The length is: %d characters.\n" as specified above, and have no other output.

## 2 String compare

`strcmp()` is a standard function from the C library that checks if two strings have the same contents. The goal of this exercise is to write a small program that checks if two input strings are identical. Unlike the real `strcmp()` function, if the strings are different you do *not* need to check which one is 'greater' than the other.

Please call your source file `myststrcmp.c`. Of course, it is forbidden to use function `strcmp()` or any similar one in your own implementation of `myststrcmp()`. An example execution looks as follows:

```
$ ./myststrcmp foo foo
The two strings are identical.
$ ./myststrcmp "" ""
The two strings are identical.
$ ./myststrcmp foo foobar
The two strings are different.
$ ./myststrcmp foo bar
The two strings are different.
$ ./myststrcmp foo ""
The two strings are different.
$
```

To enable automatic testing of your submission, your program *must* use the output "The two strings are identical.\n" or "The two strings are different.\n" as in the example, and have no other output.