

# Handy String Library

---

## Work to do

You have to write functions which will serve as a mini reusable strings library.

- **mystrings.h** will contain the headers (aka declarations) of our functions.
- **mystrings.c** will contain the definitions of all the functions we will implement.
- **tests.c** will contain the main function used to test our functions.

To start this project, let's implement the following functions in `hal.c`;

- **int str\_length ( char\* s );**
  - Returns the size of the specified string not including the final NULL character.
  - Returns 0 if ( s==NULL)
- **int str\_compare ( char\* s1 , char\* s2 );**
  - Returns 1 if the strings referred by s1 and s2 are identical character per character or if they are both NULL
  - Returns 0 otherwise.

## Hints

- When comparing strings, keep in mind that if the two parameters are strings of different length, then they can't possibly be identical.

## Testing

Write `tests.c` to ensure that your functions are working. More specifically, make sure that you cover the following tests cases;

<b>Input for str_length( char* s)</b>	<b>Return Value</b>	
String Parameter	<b>Expected</b>	<b>Observed</b>
""	0	
" "	1	
"something goes here"	19	
NULL	0	

<b>Input for str_compare ( char* s1 , char *s2)</b>		<b>Return Value</b>	
String Parameter s1	String Parameter s2	<b>Expected</b>	<b>Observed</b>
NULL	NULL	1	
NULL	"something"	0	
"something"	NULL	0	
"something"	"something"	1	
"something"	"something else"	0	
"panda"	"snake"	0	

In order to make sure that your function is comparing the contents of the strings and not their starting address, you will also call the function with the above values stored in different variables (e.g. `char str1[] = "something"; char str2[] = "something";`)