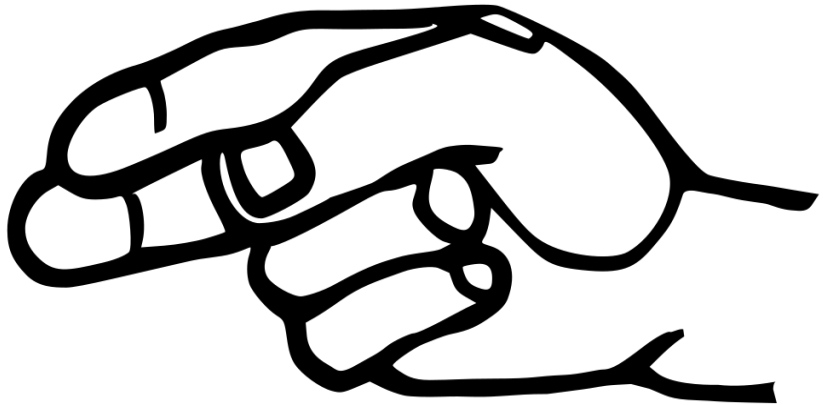
String Functions

String.h

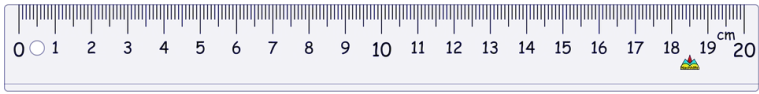
- Many functions in C are designed specifically for strings…
- ...and are placed within <string.h>



A few examples

- We will look at 3 example functions
  - strlen()
  - strcmp()
  - strcpy()

STRLEN()



- strlen() finds the length of a string
  - NOT including the trailing ‘\0’ character!

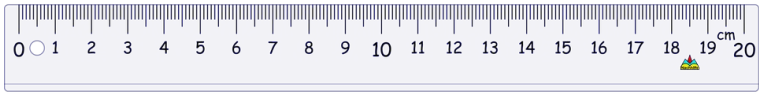
- Takes in a pointer to the first memory address and counts the characters until the first ‘\0’ character

Removing the newline character

- In order to remove this trailing newline, include the following lines of code:

```
int i = 0;
while (array_name[i] != '\0') {
        i++
}
if (i > 0 && array_name[i - 1] == '\n') {
        array_name[i - 1] = '\0';
}
```

STRLEN()



- Returns an integer
  - Can be used to pass the length of strings to functions

```
> rot13(a_string, strlen(a_string);
```

STRCMP()

- Why would this code not logically work?


```
> char str1[] = “Hello”;
> char str2[] = “Hello”;
> if (str1 == str2) {
>     printf(“They’re the same!\n”);
>}
```

STRCMP()

- In order to easily compare strings, strcmp() takes two strings as input
  - Returns **+1** if the first string is first
    - First is determined by ASCII values
  - Returns **-1** if the second string is first
  - Returns **0** if both are equal

STRCMP()

- Why would this code not logically work?


```
> char str1[] = “Hello”;
> char str2[] = “hello”;
> if (strcmp(str1, str2) == 0) {
>    printf(“They’re the same!\n”);
>}
```

STRCPY()

- Likewise, string copying does not work the same way as with primitive data types

```
> char str1[] = “Hello”;
> char str2[] = str1;
```

WILL NOT WORK!

STRCPY()

- This also doesn’t work with any kind of arrays

```
> int num_array1[] = {1, 2, 3};
> int num_array2[] = num_array1;
```

WILL NOT WORK!

strcpy()

- strcpy() takes in two pre-defined strings, and copies the second parameter into the first
  - Including any trailing ‘\0’

```
> char str1[] = “Hello”;

> char str2[];

> strcpy(str2, str1);
```

String Function Coding Challenge

- Password check
  - Have the user scan in a username, a password, and then the password again (to ensure it is legitimate).
  - Restrictions: The password MUST contain an exclamation point OR a question mark. Otherwise, continually prompt the user to enter a new password