

Tutorial 8

- Strings

Strings Review

- A string in C is an array of characters that is NULL terminated.
- For example:

```
char s1[] = {'s', 't', 'r', 'i', 'n', 'g'};  
char s2[] = {'s', 't', 'r', 'i', 'n', 'g', '\\0'};  
char s3[] = "string";  
  
// s1 is not a string  
// s2 is a valid string  
// s3 is also a valid string (it IS null terminated)
```

Useful `<string.h>` functions

```
// strlen(s) returns the length of the string s.  
// The null terminator is not counted in the length.
```

```
int strlen(const char s[])
```

```
// strcmp(s1, s2) compares s1 and s2 lexicographically.  
// If the strings are identical, it returns 0.  
// If s1 comes before s2, it returns an int < 0.  
// If s2 comes before s1, it returns an int > 0.
```

```
int strcmp(const char s1[], const char s2[])
```

Useful `<string.h>` functions

```
// strcpy(dest,src) copies the string src  
//   to the string dest.  
// effects: replaces what was in dest
```

```
char *strcpy(char *dest, const char *src)
```

```
// strcat(dest,src) copies (appends or concatenates) src  
//   to the end of dest.  
// effects: modifies dest
```

```
char *strcat(char *dest, const char *src)
```

Never put `strlen` or other $O(n)$ functions that do not have side effects in a loop condition! Beginners often make this mistake, which increases runtimes.

```
// Don't do this (strlen is called on every iteration)
for (int i=0; i<strlen(s); ++i) {

}
```

```
// Do this instead (assumes the length does not change)
int len = strlen(s);
for (int i=0; i<len; ++i) {

}
```

Exercise 1: Converting Strings

Implement the following program.

```
// This program reads some strings from input and for
// certain strings it converts them to a new string.
// Convert only the strings below
//     "computers" => "computers rule"
//     "poverty" => "poverty sucks"

// The maximum length of any string will be 80.
```

Exercise 2: Read and Print Strings

Using an array of structs, implement the following program:

```
// This program reads a bunch of strings, determines
// the length of each string and stores that
// string and its length in an array of structures.
// After all input has been read,
// it prints the strings and their length
// in the reverse order from which they were read.

// The maximum string length will be 80
// The maximum string number of strings will be 200

struct str_and_len {
    int strlen;    // track the length of the string
    char str[80+1]; // max length + 1 for null terminator
};
```