# CSC111

# Administrivia

- Assignment 6 Released --> Due Sunday March 27th
- Midterm 2 –>  next class

```
char val;
printf("type text into the keyboard followed by the enter-key to stop\n");
val = getchar();
while (val != '\n'){
      printf("%c\n", val);
      val = getchar();
}
printf("done\n");
```

**Sample run:**

```
type text into the keyboard
followed by the enter-key to stop
hi !
h
i


!
done
```

# fgetc();

```
FILE* file_handle = fopen("name.txt", "r");

if (file_handle != NULL) {
    char val;
    val = fgetc(file_handle);
    while (val != EOF){
        printf("%c\n", val);
        val = fgetc(file_handle);
    }
    fclose(file_handle);
    printf("done");
}
```

**if the file is empty,**
**Output:**
done

**if the file contains:**
hi !!
**Output:**
h
i

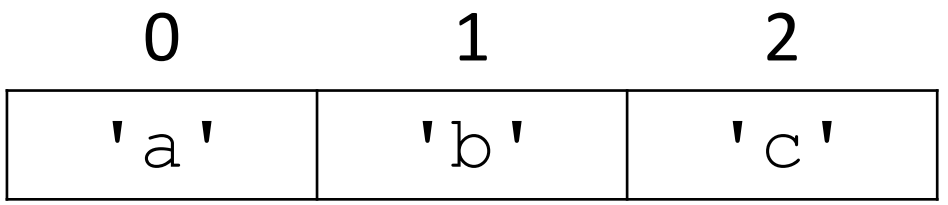
!
!
done

# ctype functions

- isalnum(char c)
  - Checks to see if a character is alphanumeric
- isalpha()
  - Checks to see if a character is alphabetic
- isdigit()
  - Checks to see if a character is a decimal digit
- isupper()/islower()
  - Checks to see if a character is upper/lower case
- Isspace()
  - Checks for a white space characters
- …and more
- Functions generally return a non 0 value if true and 0 if false.
- All accept a character as a parameter

# Strings

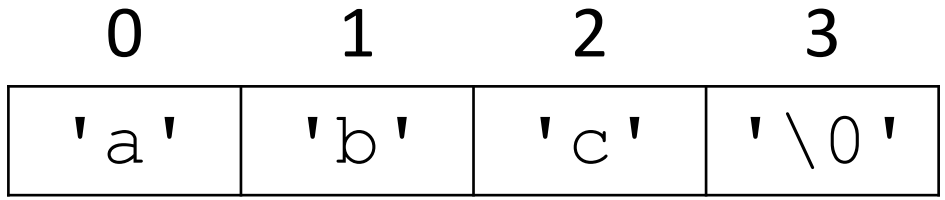
# char array…



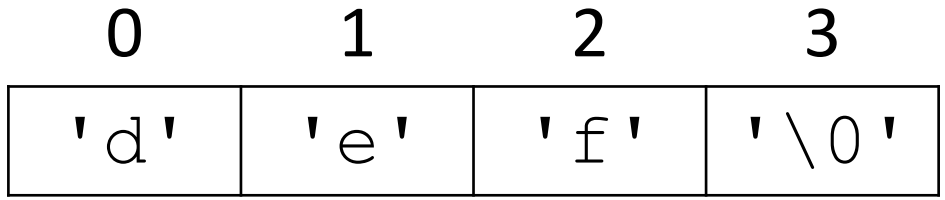

```
char my_chars[] = {'a', 'b', 'c'};

printf("value at index 0: %c\n", my_chars[0]);
printf("value at index 1: %c\n", my_chars[1]);
printf("value at index 2: %c\n", my_chars[2]);
```

# Strings = null terminated char array…

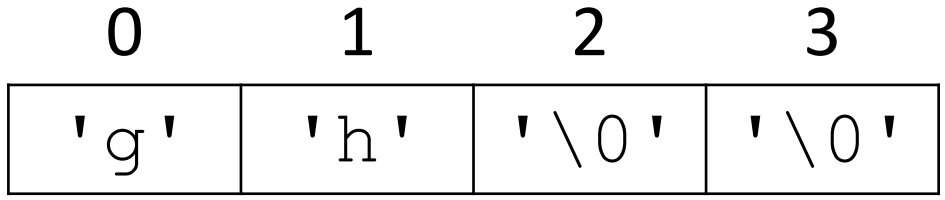
```
char my_string_1[] = {'a', 'b', 'c', '\0'};
```




```
char my_string_2[] = "def";
```




```
char my_string_2[4] = "gh";
```



```
void print_array ( char array[], int len ) {
    int i;
    for (i=0; i<len; i++) {
       printf(" %c  ", array[i]);
    }
}
```

```
void print_string (char str[]) {
    int i=0;
    while (str[i] != '\0') {
       printf("%c ", str[i++]);
    }
}
```

```
void print_string (char str[]) {

    while (*str != '\0') {
       printf("%c ", *str++);
    }
}
```

# Demo

# `string.h`

- The `string` library contains functions that operate on `string` types (null terminated `char` arrays)
- Given the prototypes and documentation of these functions, you should find them helpful in processing strings
- To use these functions in your program include the header file:

```
#include <string.h>
```

- We do not expect you to memorize the functions but given the documentation you should know how and where to use the basic ones:

https://www.tutorialspoint.com/c_standard_library/string_h.htm

# strlen

## Documentation

Purpose: Computes the length of the string str up to but not including the terminating null character.

Parameters: char str[] – a null terminated string

Returns: int, the length of the string

## Examples

```
int result;
char str1[] = "hello";
result = strlen(str1);  // result will be 5
char str2[] = "hello there!";
result = strlen(str2);  // result will be 12
```

# strcmp

Documentation

```
Purpose: compares str1 to str2 lexicographically (dictionary order)
Parameters: char str1[], char str2[] - null terminated strings
Returns: int, 0 if str1 and str2 are equal
         an int <0 if str1 comes before str2 lexicographically
         an int >0 if str1 comes after str2 lexicographically
```

Examples

```
int result;
char str1[] = "bye";
char str2[] = "bye";
char str3[] = "hello";
result = strcmp(str1, str2);  // result is 0
result = strcmp(str1, str3);  // result is a number less than 0
result = strcmp(str3, str1);  // result is a number bigger than 0
```

# strcpy

Documentation

```
Purpose: copies characters from src to dest
         up to and including the null terminator

Parameters: char dest[] – must be big enough to hold all characters
                          in src including the null terminator
            char src[] - null terminated strings

Returns: char*, a pointer to dest
```

Examples

```
char* result;
char str[] = "bye";
char dest[5];
result = strcpy(dest, str);
// dest will hold the following values: { 'b', 'y', 'e', '\0', – }
// NOTE: the – indicates a garbage value
// result is a pointer to dest
```

# Demo

# What is the output of the following code ?

```
#include <string.h>
#include <stdio.h>

int main(){

    char str_1[5] = "Hi";
    char str_2[5] = "HiHi!";
    char str_3[5] = "Bye";
    printf("str_1: %s\n", str_1);
    printf("str_2: %s\n", str_2);
    printf("str_3: %s\n", str_3);

    return 0;
}
```

# Write a function that…

```
/*
 * Purpose: count characters in src upto but not including
 *    the null terminator.
 * Parameters: char src[], a null terminated string
 * Returns: int, the count of characters
 */
int string_length(char src[]) {


}
```

NOTE: The purpose of this is to not just use the library functions, but to get an idea of how the library function is implemented

# Write a Function that…

- Design and test a function called string_reverse that takes a char array which is a null terminated string and reverses the order of the characters in the string.