STRINGS IN C

In this assignment we are going to demonstrate:

- String Input/Output/Iterating.
- ASCII value in C.
- Passing Strings to Functions.
- String handling library (<string.h>).

Part 1: String Input/Output/Iterating.

Representing strings in C:

- stored in arrays of characters.
- array can be of any length.
- end of string is indicated by a delimiter, the zero character '\0'.

String Variables:

Allocate an array of a size large enough to hold the string (plus 1 extra value for the delimiter).

Example:
char str1[15];
Examples (with initialization):
char str1[6] = "Hello";
char str2[] = "Hello";

String Input:

i. Use %s field specification in scanf to read string
   - ignores leading white space
   - reads characters until next white space encountered
   - C stores null (\0) char after last non-white space char
   - Reads into array.

   Example:
   char Name[11];
   scanf("%s",Name);

ii. use gets(char str[]) function:
   reads the next line (up to the next newline) from keyboard and stores it in the array of chars.

   Example:
   char Name[11];
   gets(Name);

String output:

i. Use %s field specification in printf to print string

Example:
char Name[11]="cs1160";
printf("%s",Name);

ii. use puts(char str[]) function:

Example:
char Name[11];
gets(Name);
puts(Name);

Iterating:

We need to know the length of the array to traverse an integer array, whereas we may use the null character in the case of string to identify the end the string and terminate the loop.

Let's see an example of counting the number of 'A' or 'a' in a string.

```
#include<stdio.h>
int main() {
   char s[45] = "C is a general-purpose computer programming";
   int i = 0;
   int count = 0;
   while (s[i] != '\0') {
      if (s[i] == 'a' || s[i] == 'A') {
         count++;
      }
      i++;
   }
   printf("The number of A/a: %d", count);
   return 0;
}
```

The number of A/a: 3

Part 2: ASCII value in C.

In C programming language, a character variable does not contain a character value itself rather the ascii value of the character variable. The ascii value represents the character variable in numbers, and each character variable is assigned with some number range from 0 to 127. The ASCII codes for the lowercase letters 'a' through 'z' are 97 to 122. On the other hand, the ASCII codes for 'A' through 'Z' are 65 to 90.

Let's see an example of counting the upper case letters

```
#include <stdio.h>
int main() {
   int i;
   int uppercount = 0;
   char w[45] = "C is A General-purpose Computer Programming";
   for (i = 0; w[i] != 0; i++) {
      if (w[i] >= 'A' && w[i] <= 'Z') {
         uppercount++;
      }
   }
   printf("The Upper: %d", uppercount);
   return 0;
}
```

The Upper: 5

Let's see an example of changes all lowercase letters to uppercase letters

```
#include <stdio.h>
int main() {
   int i;
   int uppercount = 0;
   char w[45] = "C is A Computer Programming";
   for (i = 0; w[i] != 0; i++) {
      if (w[i] >= 'a' && w[i] <= 'z') {
         w[i] = w[i] - 32;
      }
   }
   puts(w);
   return 0;
}
```

C IS A COMPUTER PROGRAMMING

Part 3: Passing Strings to Functions:

Strings can be passed to a function in a similar way as arrays.

Let's see an example of counting how many characters in a string.

```
#include <stdio.h>
int length(const char s[]) {
   int i;
   for (i = 0; s[i] != '\0'; i++);
   return i;
}
int main() {
   char st1[] = "CS 1160, Computing Fundamentals-Lab";
   int c = length(st1);
   printf("the length is : % d", c);
   return 0;
}
```

the length is : 35

Part 4: String handling library (<string.h>)

String handling library (<string.h>) has functions to

- Manipulate string data
- Search strings
- Tokenize strings
- Determine string length

Function prototype
xxx strcpy(char [] s1, const char [] s2) Copies string s2 into array s1. The value of s1 is returned.
xxx strncpy(char []s1, const char []s2, int n) Copies at most n characters of string s2 into array s1. The value of s1 is returned**.**
xxx strcat(char []s1, const char []s2) Appends string s2 to array s1. The first character of s2 overwrites the terminating null character of s1. The value of s1 is returned.
xxx strncat(char []s1, const char []s2, int n) Appends at most n characters of string s2 to array s1. The first character of s2 overwrites the terminating null character of s1. The value of s1 is returned**.**

Let's see an example of strnca.

```
#include <stdio.h>
#include <string.h>
int main() {
   char s1[50] = "Hi I am a C1160 student";
   char s2[50] = "What about you?";
   strncat(s1, s2, 29);
   printf("%s\n", s1);
   return (0);
}
```

Hi I am a C1160 studentWhat about you?

Let's see an example of strcpy.

```
#include<stdio.h>
#include<string.h>
int main() {
   char str1[] = "Hello Geeks!";
   char str2[] = "GeeksforGeeks";
   char str3[40];
   strcpy(str2, str1);
   strcpy(str3, "Copy successful");
   printf("str1: %s\nstr2: %s\nstr3: %s\n", str1, str2, str3);
   return 0;
}
```

str1: Hello Geeks!
str2: Hello Geeks!
str3: Copy successful

Simple:

1. Write a program which inputs a sentence from the user and then inputs a character ch from the user. Then it calls a function which counts the number of occurrences of the character ch in the sentence. Finally, it should print out the result of calling the function.

2. Write a program that reads a string then toggles this string (converts all uppercase letters to lowercase letters, and converts all lowercase letters to uppercase letters).

3. A palindrome is a word or sentence that reads the same forward as it does backward. Examples of these words are the following names: eye, anna and elle. write a program that inputs a word from the user and determines whether or not it is a palindrome. To do so, we need the following functions:

   Int Length(char str[])
   this function takes a string and returns its length (i.e. how many characters it is composed of)
   void reverse(char str1[], char str2[]):
   this function takes two strings and produces in the second string a sequence of characters which is a reverse of the first one. For example, the function takes the string “CS111” and produces “111SC”. Note that this function will need to use the length function to produce the reverse order.
   Int strComp(char str1[], char str2[])
   this function takes two strings and returns 1 if they are equal and 0 otherwise.