CSE115L – Programming Language I Lab

Lab – 19

String (Functions)

In this lab, we will use functions related to strings. The following examples will help you remember the syntax.

C supports a wide range of functions that manipulate NULL-terminated strings.

`strlen(s)` - Returns the length of string s.

`strcpy(s1, s2)` - Copies string s2 over string s1.

`strncpy(s1, s2)` - Copies n characters of string s2 over string s1 (without copying the NULL character).

`strcat(s1, s2)` - Concatenates string s2 at the end of string s1.

`strncat(s1, s2)` - Concatenates n characters of string s2 at the end of string s1 (without copying the NULL character).

`strcmp(s1, s2)` - Compares strings s1 and s2 alphabetically (returns negative value if s1 is alphabetically smaller than s2, 0 if they are identical and positive value if s1 is alphabetically larger than s2)

`strncmp(s1, s2)` - Compares n characters of strings s1 and s2 alphabetically

Example 1:

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

int main()
{
    char
    str1[10],str2[10],str3[20];int
    len;

    printf("Enter String
    1:");gets(str1);
    printf("Enter String
    2:");gets(str2);

    len=strlen(str1);
    printf("Length of string 1: %d\n", len);

    strcat(str1,str2);
    printf("%s\n",str1);

    strcpy(str3,str1);
    printf("%s",str3);

    return 0;
}
```

Example 2: String Comparison

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

int main()
{
   char str1[10];
   char str2[10];
   char str3[20];
   int chk;

   gets(str1);gets(str2);

   chk = strcmp(str1,str2);
   printf("%d",chk);

   if(chk == 0)
          printf("Same");


   if(chk < 0)
```

```
       printf("str1 issmaller");

    if(chk>0)
       printf("str2 issmaller");

    return 0;
}
```

Example 3: Passing string as an argument of a function and update that string within the function

```
#include <stdio.h>
#include<ctype.h>
#define SIZE 100

void changeCase(char s[])
{
    int i;

    for (i = 0; s[i] != NULL; i++)
    {
        if (isupper(s[i]))
            s[i] = tolower(s[i]);
        else if (islower(s[i]))
            s[i] = toupper(s[i]);
    }
}

void main ()
{
        char str[SIZE];

        printf ("Enter a string of length < %d : ", SIZE);
        gets(str);

        changeCase(str);
        puts(str);
}
```

Perform the following tasks.

Task 1: Write a C program to print all unique letters in an input string.

Sample input/output:
Enter a string: **Hello how are you?**
Unique letters in the input string (ignoring differences between lowercase & uppercase letters):
a, e, h, l, o, r, u, w, y

Task 2: Write a program that reads a string from user and then prints the number of times different letters appear in that string.

Sample input/output:
Enter a string: **Hello how are you?**
Frequencies of letters in the input string:
a/A: 1, e/E: 2, h/H: 2, l/L: 2, o/O: 3, r/R: 1, u/U: 1, w/W: 1, y/Y: 1,