

Title: String in C.

Objective:

The main objectives of this lab are to

- Learn about how to display String Input output, copy, lower case, concatenation with and without built in function.
- Learn about how to display String Length with and without built in function.
- Learn about hoe to Finding the Frequency of characters.

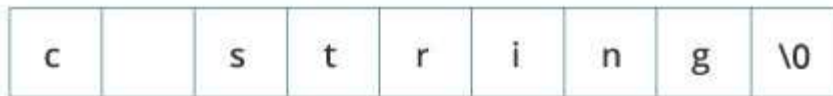
Theory:

In C programming, a string is a sequence of characters terminated with a null character `\0`.

For example:

```
char c[] = "c string";
```

When the compiler encounters a sequence of characters enclosed in the double quotation marks, it appends a null character `\0` at the end by default.



Memory Diagram

Declaration of a string

Here's how you can declare strings:

```
char s[5];
```

CSE 112

s[0]	s[1]	s[2]	s[3]	s[4]

String Declaration in C

Here, I have declared a string of 5 characters.

Initialization of strings

I can initialize strings in a number of ways.

```
char c[] = "abcd";
char c[50] = "abcd";
char c[] = {'a', 'b', 'c', 'd', '\0'};
char c[5] = {'a', 'b', 'c', 'd', '\0'};
```

c[0]	c[1]	c[2]	c[3]	c[4]
a	b	c	d	\0

String Initialization in C

Another example:

```
char c[5] = "abcde";
```

Here, I am trying to assign 6 characters (the last character is '\0') to a `char` array having 5 characters. This is bad and you should never do this.

Assigning Values to Strings

Arrays and strings are second-class citizens in C; they do not support the assignment operator once it is declared. For example,

```
char c[100];  
c = "C programming"; // Error! array type is not assignable.
```

Read String from the user

I can use the `scanf()` function to read a string.

The `scanf()` function reads the sequence of characters until it encounters whitespace (space, newline, tab, etc.).

Example 1: `scanf()` to read a string

```
#include <stdio.h>  
int main()  
{  
    char name[20];  
    printf("Enter name: ");  
    scanf("%s", name);  
    printf("Your name is %s.", name);  
    return 0;  
}
```

Output

```
Enter name: Dennis Ritchie  
Your name is Dennis.
```

Even though `Dennis Ritchie` was entered in the above program, only `"Dennis"` was stored in the `name` string. It's because there was a space after `Dennis`.

Also notice that I have used the code `name` instead of `&name` with `scanf()`.

```
scanf("%s", name);
```

This is because `name` is a `char` array, and we know that array names decay to pointers in C. Thus, the `name` in `scanf()` already points to the address of the first element in the string, which is why I don't need to use `&`.

How to read a line of text?

I can use the `fgets()` function to read a line of string. And, you can use `puts()` to display the string.

Example 2: fgets() and puts()

```
#include <stdio.h>
int main()
{
    char name[30];
    printf("Enter name: ");
    fgets(name, sizeof(name), stdin); // read string
    printf("Name: ");
    puts(name); // display string
    return 0;
}
```

Output

```
Enter name: Tom Hanks
Name: Tom Hanks
```

Here, I have used `fgets()` function to read a string from the user.

```
fgets(name, sizeof(name), stdin); // read string
```

The `sizeof(name)` results to 30. Hence, we can take a maximum of 30 characters as input which is the size of the `name` string.

To print the string, I have used `puts(name);`.

Note: The `gets()` function can also be to take input from the user. However, it is removed from the C standard.

It's because `gets()` allows you to input any length of characters. Hence, there might be a buffer overflow.

Source Code:

```
1. #include <stdio.h>
2. #include<string.h>
3.
4. int main()
5. {
6.     /// Input Output Without Built in Function.
7.
8.     // array to store string taken as input
9.     char color[20];
10.
11.     // take user input
12.     printf("Enter your favourite color: ");
13.     scanf("%s", color);
14.
15.     // printing the input value
16.     printf("Your favourite color is: %s.", color);
17.     printf("\n");
18.
19.     /// Input Output With Built in Function.
20.
21.     char name[30];
22.     printf("Enter name: ");
```

CSE 112

```

23.         gets(name);           //Function to read string from
        user.
24.         printf("Name:  ");
25.         puts(name);           //Function to display string.
26.
27.         /// Copy String Without Built in Function.
28.
29.         char s1[100],s2[100],i;
30.         printf("\nEnter string s1: ");
31.         scanf("%s",s1);
32.         for(i=0; s1[i]!='\0'; ++i)
33.         {
34.             s2[i]=s1[i];
35.         }
36.         s2[i]='\0';
37.         printf("String s2: %s\n",s2);
38.
39.         /// Copy String With Built in Function.
40.
41.         char x[] = "Happy Birthday to You";
42.         char y[25];
43.
44.         printf("\nThe string in array x is: %s", x);
45.         printf("\nThe string in array y is: %s\n",
strcpy(y,x));
46.
47.         /// Concatenate two strings without using built in
        function.
48.
49.         char str1[50], str2[50], i1, j;
50.         printf("\nEnter first string: ");
51.         scanf("%s",str1);
52.         printf("\nEnter second string: ");
53.         scanf("%s",str2);
54.         for(i1=0; str1[i1]!='\0'; ++i1);
55.         for(j=0; str2[j]!='\0'; ++j, ++i1)
56.         {
57.             str1[i1]=str2[j];
58.         }
59.         // \0 represents end of string
60.         str1[i1]='\0';
61.         printf("\nOutput: %s\n",str1);
62.
63.         /// Concatenate two strings with using built in
        function.

```

CSE 1112

```

64.
65.     char st1[] = "Happy ";
66.     char st2[] = "New Year ";
67.     printf("\nst1 = %s", st1);
68.     printf("\nst2 = %s", st2);
69.     printf("\nstrcat(st1,st2) = %s\n\n", strcat(st1,st2));
70.
71.     /// Converts string to lowercase without using built in
       function.
72.
73.     char stri[30];
74.     printf("Enter your String(Upper case):");
75.     scanf("%[^\\n]", stri);
76.     int i6 = 0;
77.     //convert capital letter string to small letter
       string
78.     while (stri[i6] != '\\0')
79.     {
80.         if (stri[i6] > 64 && stri[i6] < 91) //or
           if(stri[i6]>='A' && stri[i6]<='Z')
81.             stri[i6] += 32;
82.             i6++;
83.     }
84.     printf("Lower case String is:%s", stri);
85.
86.
87.
88.     /// Converts string to lowercase with using built in
       function.
89.
90.     char str[ ] = "I AM THE BEST";
91.
92.     // converting the given string into lowercase.
93.     printf("%s\\n",strlwr (str));
94.
95.     /// String Length without built in function
96.
97.     char st[100],i4;
98.     printf("\\nEnter a string : ");
99.     scanf("%s",st);
100.    for (i4=0; st[i4]!='\\0'; i4++);
101.    printf("The length of the string is : %d\\n\\n",i4);
102.
103.    /// String Length with built in function
104.

```

CSE 112

```
105.     char a[20]="Program";
106.     char b[20]= {'P','r','o','g','r','a','m','\0'};
107.
108.     // using the %zu format specifier to print size_t
109.     printf("Length of string a = %zu \n",strlen(a));
110.     printf("Length of string b = %zu \n",strlen(b));
111.
112.     /// Finding the Frequency of characters.
113.
114.     char string[1000], ch;
115.     int count = 0;
116.
117.     printf("\nEnter a string: ");
118.     //fgets(str, sizeof(str), stdin);
119.     //gets(string);
120.     scanf("%[^\n]",string);
121.     printf("Enter a character to find its frequency: ");
122.     scanf("%c", &ch);
123.
124.     for (int i5 = 0; string[i5] != '\0'; ++i5)
125.     {
126.         if (ch == string[i5])
127.             ++count;
128.     }
129.
130.     printf("Frequency of %c = %d", ch, count);
131.
132.     return 0;
133. }
```


Output:

```

"\\E\\EE RUET 20\\Code Blocks C\\For lab rreport\\9th.exe"
Enter your favourite color: Black
Your favourite color is: Black.

Enter name: Golpoguccho
Name: Golpoguccho

Enter string s1: masrur
String s2: masrur

The string in array x is: Happy Birthday to You
The string in array y is: Happy Birthday to You

Enter first string: Mir
Enter second string: Masrur
Output: MirMasrur

st1 = Happy
st2 = New Year
strcat(st1,st2) = Happy New Year

Enter a string : mir masrur
The length of the string is : 3

Length of string a = 7
Length of string b = 7

Enter a string: Pneumonoultramicroscopicsilicovolcanoconiosis
Enter a character to find its frequency: o
Frequency of o = 9
Process returned 0 (0x0)   execution time : 6.917 s
Press any key to continue.

```

Discussion and Conclusion:

In this program, I work on String. Firstly, I program on Input Output without and with Built in Function of String. Secondly, I program on Copy String without and with Built in Function. It takes a string from the user and copy that string to another variable. Thirdly, I program on concatenation of two strings without and with using built in function. This program takes two strings from the user and join them together. Fourthly, I program on conversion of string to lowercase without and with using built in function. Fifthly, I program on string Length without and with using built in function. Finally, I work on Finding the Frequency of characters. This program takes a string from the user and also takes which character user wan to see. Then it prints how many are they.