String

A string is an array of characters ended with a null character ('\0'). The length of string is the number of characters present excluding the null character.

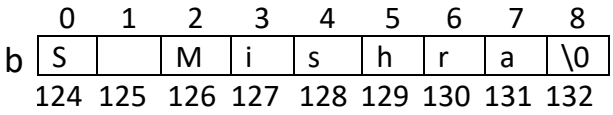
As a string is also an array, it supports index accessing.

Declaration and initialization:

syntax : char stringName[Size];

```
char a[6];      //declaration

char b[]= "S Mishra"; //declaration and initialization
```



String Input and output

Input using scanf() **:**

It can input a string up to the first white space.

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

input using gets():

gets() is a standard un-formatted input function defined in <stdio.h>. It can input a string including the spaces.

```
gets(stringanme);
```

Output with printf() :

Unlike scanf(), printf() can print the string with the spaces.

```
printf("%s",stringname);
```

Output with puts():

puts() is a standard un-formatted input function defined in <stdio.h>. It is used to print a string including the spaces.

```
puts(stringanme);
```

Program: Write a program to input your name and count the length.

```
#include<stdio.h>
int main()
     {
     char s[70];
     int i,c=0;
     printf("Enter your name:");
     gets(s);
     for(i=0; s[i]!='\0';i++)
          c++;
     printf("Name length = %d",c);
     return 0;
     }
```

What is the difference between a string and array of characters?

Programs Exercise:

(Do following programs without using string library functions)

1. Input a string. Display all characters line by line
2. Input a string. Count the characters.
3. Input a string. Count the capital letters, small letters, digits ,spaces and symbols.
4. Input a string. Count the words present in it.
5. Input a string. Convert all alphabets to upper case.
6. Input a string. Convert all alphabets to Lower case.
7. Input your name. Print in short form.
8. Input a string. Print the first word.
9. Input a string. Reverse it.
10. Input a string. Copy it to another string.
11. Input two strings. Join them.

String handling library functions

Following functions are defined in <string.h>

strlen() : returns the length of the string

```
syntax:   strlen(string);
```

strrev() : It reverses the string

```
syntax:   strrev(string);
```

strcpy() : Copies the source string to target string.

```
syntax:   strcpy(TargetString, SourceString);
```

strcat() : Concatenates the second string with first string.

```
syntax:   strcpy(String1, String2);
```

strcmp() : Compares two strings in lexicographical order. Returns

a +ve (difference), If first string lexicographically greater than second string

a -ve number(difference), If first string lexicographically less than second string

0 , if both the strings are lexicographically equal.

```
syntax:   strcmp(String1, String2);
```

Recommended other string functions to learn

strlwr(string)

strupr(string)

strcmpi(string1,string2)

strncmp(string1,string2,n)

strset(string,char)

strncpy(dest,source,n)

Program: Input a string.Input a word. Search the word in the string and print the position

```
#include<stdio.h>
#include<string.h>
int main()
     {
     char a[80],b[30],w[30];
     int i,j,loc=-1;
```

```
    printf("Enter a string:");    gets(a);
    printf("Enter a word to search:");
    scanf("%s",w);
    j=0;
    for(i = 0;i<=strlen(a);i++)
         {
         if(a[i]!=32 && a[i]!='\0')
              b[j++] = a[i];
         else
              {
              b[j] = '\0';
              if(strcmp(w,b)==0)
                   {
                   loc = i - strlen(w);
                   break;
                   }
              j = 0;
              }
         }
    if(loc==-1)
         printf("Given word is not present");
    else
         printf("The word is present in %d position",loc);
    return 0;
    }
```

Programs:

1. Input a string. Print the all the words in reverse order.
2.Input a string. Input a word. Delete the word from the string.
3. Input a string. Input a word. Locate the position of word in the string.
4. Input a string. Print the last word.
5. Input a string. Delete the last word.
6. Input a string. Delete the first word.
7. Input a string. Input a word and a position. Insert the word in given position.
8. Input a string. print the frequencies of each word in the string
9. Input a string. Print the length of each word.
10. Input your name. Print as follows
    Input : Shesha Shankar Gnanindranath Mishra
    Output : S S G Mishra
11. Input a string. Print the average length of words.