<u>String</u>

A string can be defined as a collection of characters being **terminated by null character.**

A string constant is a one-dimensional array of characters terminated by a null **('\0')** means String is a character array whose **last character is null character.**

Note: T**he null character indicates the end of the string.**

'\0' and '0' are not same. ASCII value of '\0' is 0, whereas ASCII value of '0' is 48.

<u>Declaration of String</u>

char stringname[size];

Here size represent the total number of characters that we can store in the string.

null **will not be counted in the length of the string**. It is used to show the **end of the string.**

char name[20];

The elements of the character array are stored in contiguous memory locations.

Difference between array and string

Array

Array is a data structure that holds a collection of elements having the same data types.

Array is not ended with null character by default.

String

String is a collection of characters.

The last character of the string will always be NULL character.

<u>Initilization of String</u>

<u>Compile time Initialization:</u>

char name[10]= { 'A','b','h','a','y','\0'};

char name[10]="Abhay";

char name[22]="Hello Abhay";

<u>Run Time Initialization</u>

1. **scanf()**

char name[100];

scanf("%s",name);

2. **<u>gets()</u>**

Syntax: gets(stringvariablename);

char name[100];

gets(name);

<u>Disadvantages of scanf()</u>

We cannot **input multiple string using scanf(), because it counts space as a terminator.**

char name[100]=”Hello Abhay”;

We cannot input “Hello Abhay” using scanf(). So we **overcome t**he disadvantages of string using **gets() function.**

gets()

It s is library function **present** in the **stdio.h header file** which is use to input a string during run time.

Syntax

char name[100];

gets(name);

How to print a String

For this we use a standard libaray function name **puts()**

Syntax: puts(stringvariable);

char name[100];

gets(name);

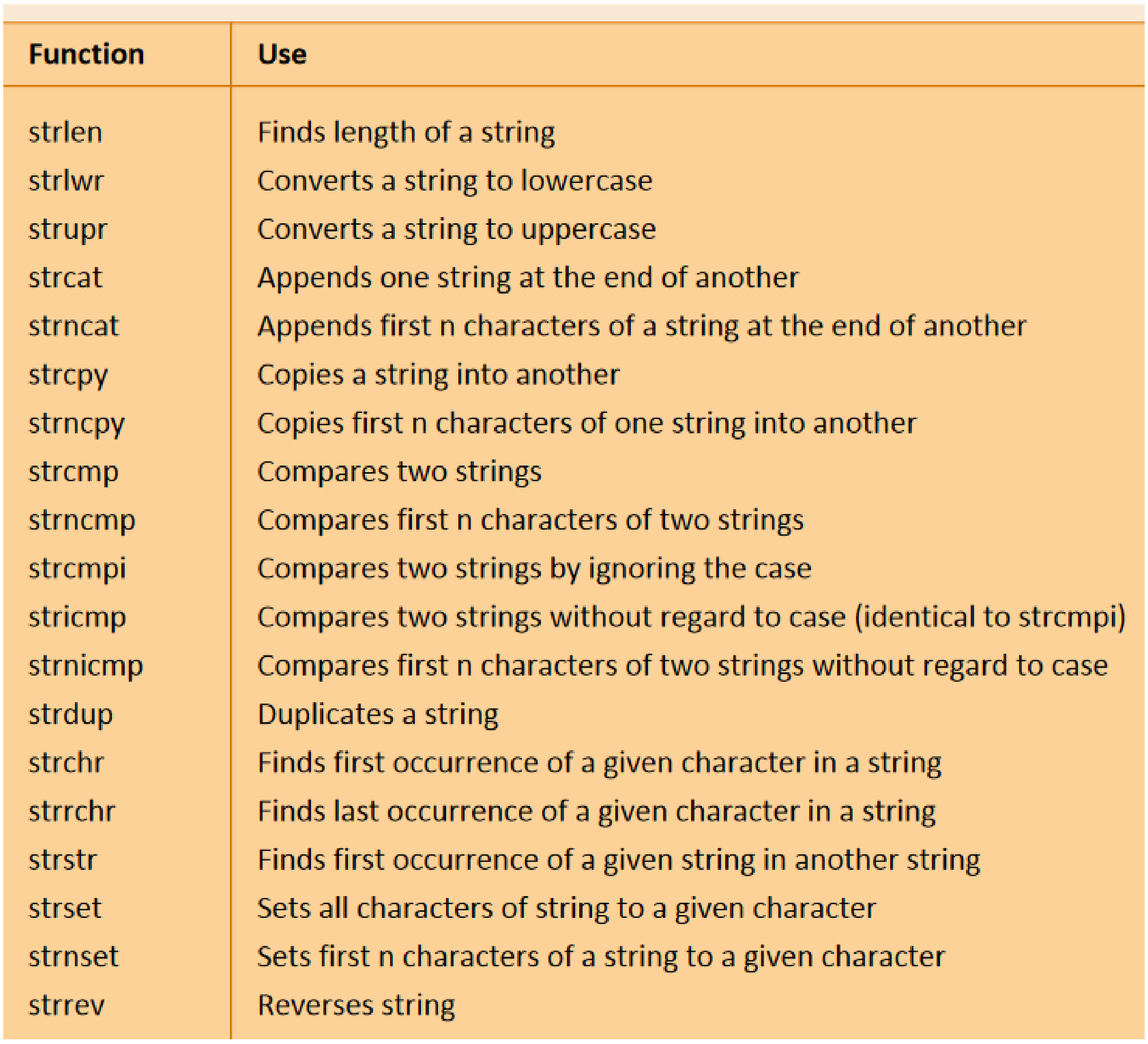
puts(name);

//Write a program in c to input a string and print it.

```
#include<stdio.h>

void main()

{

char name[20];

printf("enter the string");

gets(name);  //input the string

puts(name);  // output the string

}
```

String handling function

To deal with string we have various string handling function which are present in **“string.h”** header file.



Function	Use
strlen	Finds length of a string
strlwr	Converts a string to lowercase
strupr	Converts a string to uppercase
strcat	Appends one string at the end of another
strncat	Appends first n characters of a string at the end of another
strcpy	Copies a string into another
strncpy	Copies first n characters of one string into another
strcmp	Compares two strings
strncmp	Compares first n characters of two strings
strcmpi	Compares two strings by ignoring the case
stricmp	Compares two strings without regard to case (identical to strcmpi)
strnicmp	Compares first n characters of two strings without regard to case
strdup	Duplicates a string
strchr	Finds first occurrence of a given character in a string
strrchr	Finds last occurrence of a given character in a string
strstr	Finds first occurrence of a given string in another string
strset	Sets all characters of string to a given character
strnset	Sets first n characters of a string to a given character
strrev	Reverses string

<u>String library functions are as follows:</u>

strlen(): it is used to calculate the length of the string.

strcpy(): it is used to copy a string into another string.

strrev(): it is used to reverse a string()

strcat(): it is used to concatenate(join) two string together.

strcmp(): it is used to compare the string.

//write a program to calculate the length of string with using string library function

```
#include<stdio.h>
#include<string.h>   // string header file for strlen function
void main()
{
char name[20];
int len;
printf("enter name\n");
gets(name);
len=strlen(name);
printf("length of the string is \t%d",len);
}
```

//write a program to calculate the length of string without using string library function

```
#include<stdio.h>
void main()
{
char name[20];
int len=0,i;
printf("enter name\n");
gets(name);
for(i=0; name[i]!='\0';i++)
len++;
printf("length of the string is \t%d",len);
}
```

//string copy

Syntax:

strcpy(target string, source string)

Here content of source string will copy to target string.

//write a program to copy a string into another string with using string library function

```
#include<stdio.h>

#include<string.h>

void main()

{

char source[20],target[20];

printf("enter string\n");

gets(source);

strcpy(target,source);

printf("The source string is \t");

puts(source);

printf("The target string is \t");

puts(target);

}
```

//write a program to copy a string into another string without using string library function

```
#include<stdio.h>

void main()

{

char source[20],target[20];

int i;

printf("enter string\n");

gets(source);

for(i=0;source[i]!='\0';i++)

{

target[i]=source[i];

}
target[i]='\0';
printf("The source string is \t");
puts(source);
printf("The target string is \t");
puts(target);
}
```

<u>strcat()</u>

strcat(target string, source string);

After concatenation: targetstringsourcestring

//Write a c program to concatenate two string with using library function

```
#include<stdio.h>
#include<string.h>
void main()
{
char first[20],second[20];
printf("enter ist string\n");
gets(first);
printf("enter 2nd string\n");
gets(second);
strcat(first,second);
printf("after concatenation\n");
printf("ist string\t");
puts(first);
printf("2nd string\t");
puts(second);
}
```

//Write a c program to concatenate two strings without using library function

```
#include<stdio.h>
void main()
{
char first[20],second[20];
int len=0,i;
printf("enter ist string\n");
gets(first);
printf("enter 2nd string\n");
gets(second);
for(i=0;first[i]!='\0';i++)
{
len++;
}
for(i=0;second[i]!='\0';i++)
{
first[len+i]=second[i];
}
first[len+i]='\0';
printf("after concatenation\n");
printf("ist string\t");
puts(first);
printf("2nd string\t");
puts(second);
}
```

strcmp()

strcmp(target string,source string)

strcmp function return the difference between the ASCII value of first mismatch characters. It **can return +1,-1,0 (in codeblocks), b**ut in TurboC++ compiler it returns the difference of ascii value between target string and source string.

```
int l=strcmp("hello","hello everyone");
```

Output: -1;

//Write a c program to check whether two strings are identical or not using library.

```
#include<stdio.h>
#include<string.h>
void main()
{
char first[20],second[20];
int d;
printf("enter ist string\t");
gets(first);
printf("enter second string\t");
gets(second);
d=strcmp(first,second);
if(d==0)
printf("identical");
else
printf("not identical");
}
```

//Write a c program to check whether two strings are identical or not without using library funciton

```
#include<stdio.h>
void main()
{
char first[20],second[20];
int d=0,i,a,b;
printf("enter ist string\t");
gets(first);
printf("enter second string\t");
gets(second);
a=strlen(first);
b=strlen(second);
if(a!=b)
d=1;
else
{
   for(i=0;first[i]!='\0'&& second[i]!='\0';i++)
{
if(first[i]!=second[i])
{
d=first[i]-second[i];
break;
}
}
}
if (d==0)
printf("identical");
else
printf("not identical");
}
```

strrev()

Syntax:

strrev(string_name);

//write a program to reverse a string with using string library function

```
#include<stdio.h>
#include<string.h>
void main()
{
char name[20];
printf("enter string\n");
gets(name);
strrev(name);
printf("after reverse \n");
puts(name);
}
```

//Write a program in C to reverse a string by using pointer.

```
#include <stdio.h>
#include <string.h>
void reverseString(char* str)
{
   int l, i;
   char *begin, *end, ch;
      l = strlen(str);
   begin = str;
   end = str + l - 1;
   for (i = 0; i < l/2; i++)
   {
      ch = *end;
       *end = *begin;
      *begin = ch;
      begin++;
      end--;
   }
}
int main()
{
   char str[100] ;
   printf("Enter a string:\n");
   gets(str);
   reverseString(str);
   printf("Reverse of the string: %s\n", str);
   return 0;
}
```

//Write a c program to check whether given string is palindrome or not using library function.

```
#include<stdio.h>
#include<string.h>
void main()
{
int d;
char first[20],second[20];
printf("enter string\t");
gets(first);
strcpy(second,first);
strrev(second);
d=strcmp(first,second);
if(d==0)
printf("palindrome");
else
printf("not palindrome");
}
```

//Write a c program to check whether given string is palindrome or not without using library function.

```
#include <stdio.h>
#include <string.h>
int ispalin(char* str)
{
   int palin=0;
   int l, i;
   char *begin, *end, ch;
      l = strlen(str);
   begin = str;
```

```
    end = str + l - 1;
    for (i = 0; i < l/2; i++)
    {
        if(*begin==*end)
        {
        begin++;
        end--;
        }
        else
            return 1;
    }
    return palin;
}

int main()
{
    int p;
    char str[100] ;
    printf("Enter a string:\n");
    gets(str);
    p=ispalin(str);
    if(p)
        printf("not palindrome");
    else
         printf("palindrome");
    return 0;
}
```

//Write a program to rearrange a list of names in ascending order.

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

```
main(){
  int i,j,n;
  char str[100][100],s[100];
  printf("Enter number of names :");
  scanf("%d",&n);
  printf("Enter names in any order:");
  for(i=0;i<n;i++){
    scanf("%s",str[i]);
  }
  for(i=0;i<n;i++)
  {
    for(j=i+1;j<n;j++)
     {
      if(strcmp(str[i],str[j])>0)
      {
        strcpy(s,str[i]);
        strcpy(str[i],str[j]);
        strcpy(str[j],s);
      }
    }
  }
  printf(" The sorted order of names are: ");
  for(i=0;i<n;i++)
   {
    printf("\n%s",str[i]);
  }
  return 0;
}
```

//Write a c program to count total no of uppercase letter, lowercase letter , space , digits and words in a string.

```
void main()
{
char str[100];
int i,u=0,l=0,s=0,d=0,sp=0;
printf("enter string\t");
gets(str);
for(i=0; str[i] !='\0' ;i++)
{
if(str[i] >=65  && str[i]<90)
u++;
else if(str[i]==32)
s=s+1;
else if(str[i]>=97 && str[i]<=122)
l=l+1;
else if(str[i]>=48 && str[i]<=57)
d=d+1;
else
sp=sp+1;
}
printf("\nnumber of uppercase %d",u);
printf("\nnumber of lowercase %d",l);
printf("\nnumber of digit %d",d);
printf("\nnumber of special character %d",sp);
printf("\nnumber of space %d",s);
}
```

Question Bank

2016-17(CS-201)

1. Write a program to rearrange a list of names in ascending order. 10

2017-18(RCS-201)

1. Write a program in C to reverse a string by using pointer. 7

2018-19(KCS-101)

1. Write short notes on following:

   (i) Enumerated data type

   (ii) String 10

2018-19(KCS-201)

1. Explain the significance of null character in string. 2

2022-23(BCS-201)

1 Discuss the following string functions in C with suitable code snippet:

(i) strrev

(ii) strcmp

(iii) strlen

(iv) strcpy 7