

## # String

String is a sequence of character that is treated as a single data item.

In C programming a string is stored as a null terminated character array i.e after the last useable character there is a null value represented by \0.

String are often used to make meaningful & readable program. The common operation performed in character (string) are :-

- (1) Read & write string
- (2) Combining string together.
- (3) Copy one string to another
- (4) Comparing string with each other
- (5) Extracting a portion of string.

### → Declaring & Initializing a string.

#### Syntax

Char ~~name~~ string name [size];

(2) Char name [10] = "Santosh";

char name [10] = { 's', 'a', 'n', 't', 'o', 's', 'h', '\0' };

→ Ways to take input in string

(1)

```
printf ("In Enter name : ");
for (i=0; i<n; i++)
    {
        scanf ("%c", &name[i]);
    }
```

(2)

```
printf ("In Enter name : ");
gets (name);
```

(3)

```
printf ("In Enter your name : ");
scanf ("%s", name);
```

## # Arithmetic Operation on character

C allows us to manipulate the characters in the same way as they do with number. Whenever a character constant or character variable is used in an expression it is automatically converted into integer value by the system.

(a)

```
char n='a';
printf ("%d", n);
↓
O/P = 97
```

(b)  $\text{char } n = 'a' - 1;$   
 $\text{printf} ("%d", n);$   
↓  
 $O/P = 96$

# Reading a line of text :-

(a) By using string handling function :-

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
void main()
{
    char name[100];
    clrscr();
    printf("Enter a word:");
    gets(name);
    puts(name);
    getch();
}
```

(b) Without using string handling function

```
#include <stdio.h>
#include <conio.h>
void main()
{
    char name[100];
    int i;
    clrscr();
    printf("Enter word:");
    for (i=0; i<100; i++)
    {
        scanf("%c", &name[i]);
        if (name[i] == '\n')
            break;
    }
}
```

name[i] = '0';

break;

{

// To print the string

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

{

    printf ("%c", name[i]);

g

     getch();

g

# String handling function :-

C library support a large number of string handling function that allows us to carry out many string manipulation. It requires a header file known as `<string.h>`.

(2) `strlen()` function (String length) :-

This functions return the length or total number of characters in a string.

Syntax

`strlen(string name)`

Example:- `#include <stdio.h>`

`#include <conio.h>`

`#include <string.h>`

`void main()`

{

```
int length;
char str[50];
clrscr();
printf("In Enter the string : ");
gets(str);
length = strlen(str);
printf("In length of %s is %d", str, length);
getch();
```

## 2) strrev() function :- (String reverse)

This function is used to reverse a string.

### Syntax

```
strrev(stringname);
```

### Example -

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
void main()
{
```

```
char str[50];
```

```
clrscr();
```

```
printf("In Enter a string : ");
```

```
gets(str);
```

```
strrev(str);
```

```
printf("In Reverse of %s is %d", str, length);
```

```
getch();
```

```
}
```

3) strcpy() function :- (String copy)  
 This function is used to copy a string to another string.

### Syntax

strcpy(destination, source);

Example:-

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
void main()
{
    char source[10] = "C Program",
        destination[50];
    clrscr();
    strcpy(destination, source);
    printf("Source String = %s\n", source);
    printf("Destination String = %s\n",
        destination);
    getch();
}
```

4) strcmp() function :- (String compare) :-  
 strcmp function compares two strings & return 0 if they are identical.  
 The strcmp() function takes two strings & return an integer.

# Return value from strcmp()

Return Value      Remarks

0

negative

if both strings are identical (equal)

if the ASCII value of first un-

matched character is less than 2<sup>nd</sup>

if the ASCII value of first un-

matched character is greater

than second.

Positive

Example:-

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

Void main()
{
    char str1[] = "abcd", str2[] = "abCd",
    str3[] = "abcd";
    int result;
    result = strcmp(str1, str2);
    printf("strcmp(str1,str2)=%d\n", result);
    result = strcmp(str1, str3);
    printf("In strcmp(str1,str3)= %d\n", result);
    getch();
}
```

Output

```
strcmp(str1,str2)=32
strcmp(str1,str3)=0
```

⑥.

to upper () function :-

Function `to upper ()` takes a single argument in the integer form & returns a value of type `int`.

Syntax

```
int to upper (int arg);
```

Example:-

```
#include <stdio.h>
#include <cctype.h>
#include <conio.h>
void main()
{
```

char c;

c = 'm';

```
printf("%c → %c", c, toupper(c));
```

It displays the same argument passed  
if other characters than

c = 'D';

```
printf("%c → %c", c, toupper(c));
```

c = 'g';

```
printf("%c → %c", c, toupper(c));
```

getch();

Output

m → M

D → D

g → G

### (6) tolower( ) functions :-

Function tolower( ) takes a single argument in the integer form & returns a value of type int. It changes a single character to the lower alphabet.

Syntax

```
int tolower(int arg);
```

Example:-

```
#include <stdio.h>
#include <conio.h>
#include <ctype.h>
void main()
{
    char c;
    c = 'M';
    printf("%c -> %c", c, tolower(c));
}
```

It displays the same argument passed, if other characters than

```
c = 'd';
printf("%c -> %c", c, tolower(c));
c = 'g';
printf("%c -> %c", c, tolower(c));
getch();
```

Output

M -> m

d -> d

g -> g

7 strupr () function :- (String upper)  
It convert lower alphabet string  
into upper alphabet.

Syntax

strupr (string name);

Example:-

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
void main()
{
    char string[100];
    clrscr();
    printf("Enter a string : ");
    gets(string);
    printf("String = %s",strupr(string));
    getch();
}
```

Output = Enter string = "go" be  
input,

Output = G20

⑧ strlower() function :- (String lower)

It convert upper alphabet  
string into lower alphabet.

Syntax

strlower(string name);

Example:-

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
void main()
{
```

char str[100];

clrscr();

printf("Enter a string : ");

`gets(str);`

`printf("String in lower case = %s", strtok(str));`

`getch();`

Input `str[100] = " HELLO"`

Output

`hello`

### (9) `Strcat()` function :- String combining.

The function `strcat()` concatenates two strings. In C programming, `strcat()` concatenates (joins) two strings. The `strcat()` function is defined in `<string.h>` header file.

Syntax

Example - `strcat(destination, source)`

`#include <conio.h>`

`#include <string.h>`

`#include <stdio.h>`

`void main()`

`{`

`char str1[ ]= "This is ", str2[ ]=`  
`" programiz.com";`

`// Concatenates str1 and str2 and`

`resultant string is stored`

`strcat(str1, str2);`

`puts(str1);`

`getch();`

Output

`→ This is programiz.com`