

UNIT-3 (STRINGS) (RAM GOPAL GUPTA- <http://ramgopalgupta.com/>)

Definition:

String is a character array terminated by null character (\0) or we can say, a string is a sequence of characters terminated with a null character \0.

The difference between a character array and a string is the string is terminated with a special character '\0'.

Declaration:

Declaring a string is as simple as declaring a one dimensional array. Below is the basic syntax for declaring a string.

```
char str_var[size];
```

In the above syntax str_var is a string variable and size is used define the length of the string, i.e. the number of characters strings will store. Please keep in mind that there is an extra terminating character which is the Null character ('\0') used to indicate termination of string which differs strings from normal character arrays.

Initialization of strings:

A string can be initialized in different ways. I will explain this with the help of an example.

1. `char str_var[] = "SMSVARANASI";`
2. `char str_var[40] = " SMSVARANASI ";`
3. `char str_var[] = {'S','M','S','V','A','R','A','N','A','S','I','\0'};`
4. `char str_var[12] = {'S','M','S','V','A','R','A','N','A','S','I','\0'};`

Following is the memory presentation of the string in C language:

	str_var[0]	str_var[1]	str_var[2]	str_var[3]	str_var[4]	str_var[5]	str_var[6]	str_var[7]	str_var[8]	str_var[9]	str_var[10]	str_var[11]
Index	0	1	2	3	4	5	6	7	8	9	10	11
Value	S	M	S	V	A	R	A	N	A	S	I	\0
Address	1005	1006	1007	1008	1009	1010	1011	1012	1013	1014	1015	1016

```
char str_var[] = "SMSVARANASI";
```

You do not place the null character at the end of a string constant. The C compiler automatically places the '\0' at the end of the string when it initializes the array.

TRY THESE CODES:

Example Code 1: -

```
#include <stdio.h>
int main () {
    char str_var[12] = {'S','M','S','V','A','R','A','N','A','S','T','\0'};
    printf("String in str_var: %s\n", str_var );
    return 0;
}
```

Example Code 2: -

```
// C program to illustrate strings
#include<stdio.h>
int main()
{
    // declare and initialize string
    char str_var[] = "SMSVARANASI";
    // print string
    printf("%s",str_var);
    return 0;
}
```

Example Code 3: -

```
// C program to read strings
#include<stdio.h>
int main()
{
    // declaring string
    char str_var[50];
    // reading string
    scanf("%s",str_var);
    // print string
    printf("%s",str_var);
    return 0;
}
```

Example Code 4: -

```
// C program to illustrate how to pass string to functions
#include<stdio.h>
void printStr(char str[])
{
    printf("String is : %s",str);
}
int main()
{
    // declare and initialize string
    char str_var[] = "SMSVARANASI";
    // print string by passing string to a different function
    printStr(str_var);
    return 0;
}
```

Passing strings to function:

As strings are character arrays, so we can pass strings to function in a same way we pass an array to a function.
←This is a sample program to do this.

STRING LIBRARY FUNCTIONS

Sr.No.	Function & Purpose
1	strcpy(s1, s2); Copies string s2 into string s1.
2	strcat(s1, s2); Concatenates string s2 onto the end of string s1.
3	strlen(s1); Returns the length of string s1.
4	strcmp(s1, s2); Returns 0 if s1 and s2 are the same; less than 0 if s1<s2; greater than 0 if s1>s2.
5	strchr(s1, ch); Returns a pointer to the first occurrence of character ch in string s1.
6	strstr(s1, s2); Returns a pointer to the first occurrence of string s2 in string s1.