Dr. Deepak Verma
Department of Computer Science
Sri Jai Narain Misra PG College, Lucknow

Topic : Strings in C Programming (For B.Sc. II sem.)

1. Define C string? How to declare and initialize C strings with an example?

Ans: C Strings:-

In C language a string is group of characters (or) array of characters, which is terminated by delimiter \0 (null). Thus, C uses variable-length delimited strings in programs.

Declaring Strings:-

C does not support string as a data type. It allows us to represent strings as character arrays. In C, a string variable is any valid C variable name and is always declared as an array of characters.

Syntax:- char string_name[size];

The size determines the number of characters in the string name.

Ex:- char city[10];

char name[30];

Initializing strings:-

There are several methods to initialize values for string variables.

Ex:- char str[6]="HELLO";

H	E	L	L	O	\0

Ex:- char month[]="JANUARY";

J	A	N	U	A	R	Y	\0

Ex:- char city[8]="NEWYORK";

char city[8]={„N","E","W","Y","O","R","K","\0"};

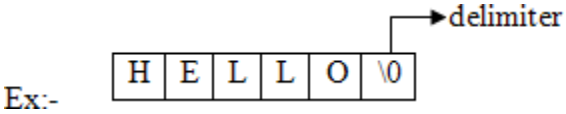
The string city size is 8 but it contains 7 characters and one character space is for NULL

terminator.

Storing strings in memory:-

In C a string is stored in an array of characters and terminated by \0 (null).

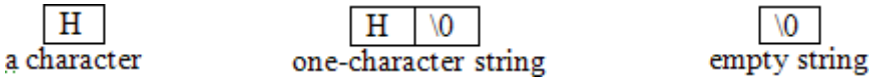
Ex:-



A string is stored in array, the name of the string is a pointer to the beginning of the string.

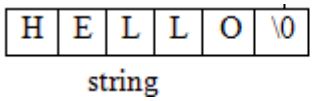
The character requires only one memory location.

If we use one-character string it requires two locations. The difference is shown below,

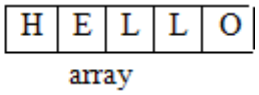


The difference between array and string is shown below,

Ex:-

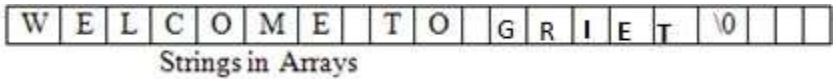


Ex:-



Because strings are variable-length structure, we must provide enough room for maximum-length string to store and one byte for delimiter.

Ex:- char str[20];



Why do we need null?

A string is not a datatype but a data structure. String implementation is logical not physical. The physical structure is array in which the string is stored. The string is variable-length, so we need to identify logical end of data in that physical structure.

String constant (or) Literal:-

String constant is a sequence of characters enclosed in double quotes. When string constants are used in C program, it automatically initializes a null at end of string.

Ex:- "Hello" "Welcome" "Welcome to C Lab"

2. Explain about the string Input/ Output functions with example?

Ans: Reading and Writing strings:-

C language provides several string handling functions for input and output.

String Input/Output Functions:-

C provides two basic methods to read and write strings. Using formatted input/output functions and using a special set of functions.

Reading strings from terminal:-

(a) **formatted input function**:- scanf can be used with %s format specification to read a string.

Ex:- char name[10];

scanf("%s",name);

Here don"t use „&" because name of string is a pointer to array. The problem with scanf is that it terminates its input on the first white space it finds.

Ex:- NEW YORK

Reads only NEW (from above example).

(b) **Unformatted input functions**:-

(1) **getchar()**:- It is used to read a single character from keyboard. Using this function repeatedly we may read entire line of text

Ex:- char ch="z";

ch=getchar();

(2) **gets()**:- It is more convenient method of reading a string of text including blank spaces.

Ex:- char line[100];

gets(line);

Writing strings on to the screen:-

(1) **Using formatted output functions**:- printf with %s format specifier we can print strings in different formats on to screen.

Ex:- char name[10];

printf("%s",name);

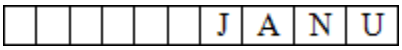
Ex:- char name[10];

printf("%0.4",name);

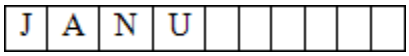
J	A	N	U

/* If name is JANUARY prints only 4 characters ie., JANU */

Printf("%10.4s",name);



printf("%-10.4s",name);



(2) **Using unformatted output functions**:-

(a) **putchar()**:- It is used to print a character on the screen.
Ex:- putchar(ch);

(b) **puts()**:- It is used to print strings including blank spaces.
Ex:- char line[15]="Welcome to lab";

puts(line);

3. Explain about the following string handling functions with example programs.

(i) strlen (ii) strcpy (iii) strcmp (iv) strcat

Ans:

C supports a number of string handling functions. All of these built-in functions are aimed at performing various operations on strings and they are defined in the header file **string.h**.

(i). strlen( )

This function is used to find the length of the string excluding the NULL character. In other words, this function is used to count the number of characters in a string. Its syntax is as follows:

```
Int strlen(string);
```

Example: char str1[ ] = "WELCOME";

int n;

n = strlen(str1);

/* A program to calculate length of string by using strlen() function*/

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

```
char string1[50];
int length;
printf("\n Enter any string:");
gets(string1);
length=strlen(string1);
printf("\n The length of string=%d",length);
}
```

(ii). strcpy( )

This function is used to copy one string to the other. Its syntax is as follows:

```
strcpy(string1,string2);
```

where string1 and string2 are one-dimensional character arrays.

This function copies the content of string2 to string1.

E.g., string1 contains master and string2 contains madam, then string1 holds madam after execution of the strcpy (string1,string2) function.

Example:

```
char str1[ ] = "WELCOME";
char str2[ ] ="HELLO";
strcpy(str1,str2);
```

/* A program to copy one string to another using strcpy() function */

```
#include<stdio.h>
#include<string.h>
main()
{
char string1[30],string2[30];
printf("\n Enter first string:");
```

```
gets(string1);

printf("\n Enter second string:");

gets(string2);

strcpy(string1,string2);

printf("\n First string=%s",string1);

printf("\n Second string=%s",string2);

}
```

(iii). strcmp ( )

This function compares two strings character by character (ASCII comparison) and returns one of three values {-1,0,1}. The numeric difference is „0" if strings are equal .If it is negative string1 is alphabetically above string2 .If it is positive string2 is alphabetically above string1.

Its syntax is as follows:

```
Int strcmp(string1,string2);
```

Example:

```
char str1[ ] = "ROM";

char str2[ ] ="RAM";

strcmp(str1,str2);

(or)

strcmp("ROM","RAM");
```

/* A program to compare two strings <u>using strcmp()</u> function */

```
#include<stdio.h>

#include<string.h>

main()

{

char string1[30],string2[15];

int x;

printf("\n Enter first string:");
```

gets(string1);

printf("\n Enter second string:");

gets(string2);

x=**strcmp(string1,string2);**

if(x==0)

printf("\n Both strings are equal");

else if(x>0)

printf("\n First string is bigger");

else

printf("\n Second string is bigger");

}

(iv). strcat ( )

This function is used to concatenate two strings. i.e., it appends one string at the end of the specified string. Its syntax as follows:

```
strcat(string1,string2);
```

where string1 and string2 are one-dimensional character arrays.

This function joins two strings together. In other words, it adds the string2 to string1 and the string1 contains the final concatenated string. E.g., string1 contains **prog** and string2 contains **ram**, then string1 holds **program** after execution of the strcat() function.

Example: char str1[10] = "VERY";

char str2[5] ="GOOD";

strcat(str1,str2);

/* A program to concatenate one string with another <u>using strcat() function</u>*/

#include<stdio.h>

#include<string.h>

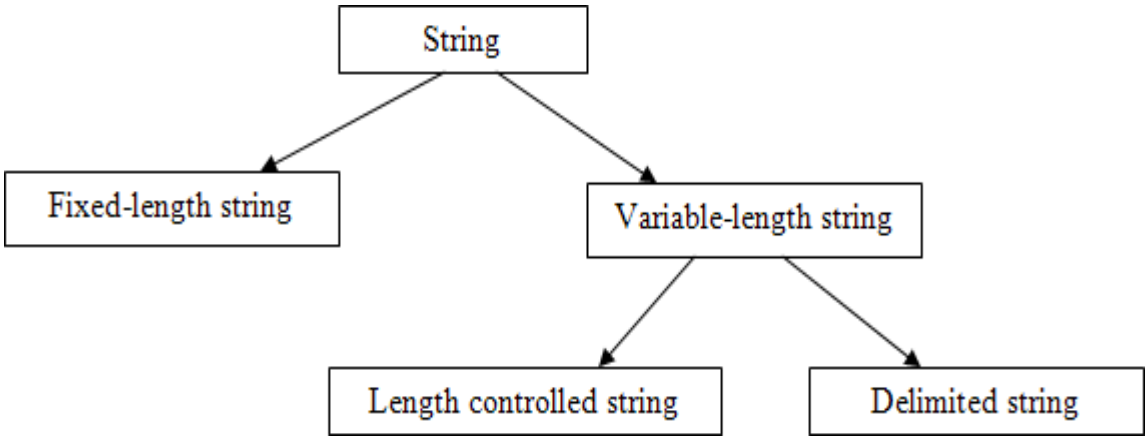
main()

```
{
char string1[30],string2[15];
printf("\n Enter first string:");
gets(string1);
printf("\n Enter second string:");
gets(string2);
strcat(string1,string2);
printf("\n Concatenated string=%s",string1);
}
```

4. Write about storage representation of fixed and variable length format strings with example?

Ans: String concepts:-

In general a string is a series of characters (or) a group of characters. While implementation of strings, a string created in pascal differs from a string created in C language.



1. **Fixed-length strings:**
   When implementing fixed-length strings, the size of the variable is fixed. If we make it too small we can‟t store, if we make it too big, then waste of memory. And another problem is we can‟t differentiate data (characters) from non-data (spaces, null etc).

2. **Variable-length string:**

The solution is creating strings in variable size; so that it can expand and contract to accommodate data. Two common techniques used,

(a) Length controlled strings:

These strings added a count which specifies the number of characters in the string.

Ex:-

5	H	E	L	L	O

(b) Delimited strings:

Another technique is using a delimiter at the end of strings. The most common delimiter is the ASCII null character (\0). Ex:-

H	E	L	L	O	\0

11. How can we declare and initialize Array of strings in C? Write a program to read and display array of strings.

Ans: We have array of integers, array of floating point numbers, etc.. similarly we have array of strings also.

Collection of strings is represented using array of strings.

Declaration:-

Char arr[row][col];

where,

arr - name of the array

row - represents number of strings

col - represents size of each string

Initialization:-

char arr[row][col] = { list of strings };

Example:-

char city[5][10] = { “DELHI”, “CHENNAI”, “BANGALORE”, “HYDERABAD”,

“MUMBAI” };

D	E	L	H	I	\0				
C	H	E	N	N	A	I	\0		
B	A	N	G	A	L	O	R	E	\0
H	Y	D	E	R	A	B	A	D	\0
M	U	M	B	A	I	\0			

In the above storage representation memory is wasted due to the fixed length for all strings.

**********Thank You**********