RV College of Engineering®

Go, change the world

PRINCIPLES OF PROGRAMMING IN C (22CS23)

UNIT III

Strings

Strings

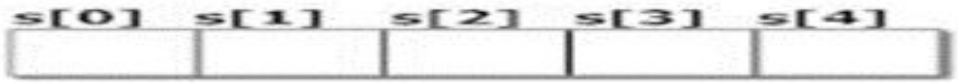
Strings

- In C programming, array of characters are called strings. A string is terminated by null character /0. For example:
- "c string tutorial“

- Here, "c string tutorial" is a string. When, compiler encounters strings, it appends null character at the end of string.

Declaration and initializing Strings Variables

- Strings are declared in C in similar manner as arrays. Only difference is that, strings are of char type.
- char s[5];

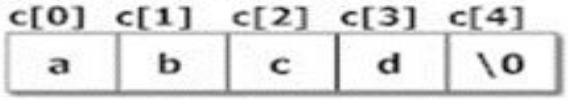


- Strings can also be declared using pointer.
- char *p

Declaration and initializing Strings Variables

In C, string can be initialized in different number of ways.

```
char c[]="abcd";
    OR,
char c[5]="abcd";
    OR,
char c[]={'a','b','c','d','\0'};
    OR;
char c[5]={'a','b','c','d','\0'};
```



String can also be initialized using pointers

```
char *c="abcd";
```

Reading Strings from terminal

Reading words from user:

```
char c[20];
scanf("%s",c);
```

String variable c can only take a word. It is because when white space is encountered, the scanf() function terminates.

Reading Strings from terminal

Write a C program to illustrate how to read string from terminal.

```
#include <stdio.h>
int main(){
    char name[20];
    printf("Enter name: ");
    scanf("%s",name);
    printf("Your name is %s.",name);
    return 0;
}
```

Output

```
Enter name: Dennis Ritchie
Your name is Dennis.
```

Here, program will ignore Ritchie because, scanf() function takes only string before the white space.

Reading Strings from terminal

Write a C program to illustrate how to read string from terminal.

```
#include <stdio.h>
int main(){
    char name[20];
    printf("Enter name: ");
    scanf("%s",name);
    printf("Your name is %s.",name);
    return 0;
}
```

Output

```
Enter name: Dennis Ritchie
Your name is Dennis.
```

Here, program will ignore Ritchie because, scanf() function takes only string before the white space.

Reading String with spaces by using scanf

Note that –

scanf with %s accepts only String which does not contain whitespaces (blanks/space)

It Reads wide verity of Characters including blank

Syntax :

```
scanf("%[\^n]", name );
```

Example :

```
void main()
{
char name[100];
printf("\nEnter the name : ");
scanf("%[\^n]s",name);
printf ("\nName of Student : %s ",name);
}
```

Output of this Block :

Enter the Name : RV CE

Name of Student : RV CE

Reading a line of text:

C program to read line of text manually.

```
#include <stdio.h>
int main(){
   char name[30],ch;
   int i=0;
   printf("Enter name: ");
   while(ch!='\n')    // terminates if user hit enter
   {
      ch=getchar();
      name[i]=ch;
      i++;
   }
   name[i]='\0';       // inserting null character at end
   printf("Name: %s",name);
   return 0;
}
```

This process to take string is tedious. There are predefined functions gets() and puts() in C language to read and display string respectively.

Writing Strings to screen

Using printf function:

Printf function can be used to print a string on to the terminal. For example:

printf("Name: %s",name);We can also specify the precision with which the array is displayed. For example:

printf("Name: %10.4s",name);

The above line indicates that the first four characters are to be printed in a field width of 10 columns. So generalizing

printf("Name: %*.*s",w,d,name);

prints the first d characters of the string in the field width of w.

Arithmetic operations on characters

Characters in C can be used just like integers when used with arithmetic operators.

Whenever a character constant or character variable is used in an expression, it is automatically converted into an integer value by the system. The integer value depends on the local character set of the system.

Examples :

ASCII value of : ‘a’ is 97

ASCII value of : ‘z’ is 121

Arithmetic operations on characters

Possible Ways of Manipulation:

Way 1: Displays ASCII value[Note that %d in Printf]

```
char x = 'a';
printf("%d",x); // Display Result = 97
```

Way 2 : Displays Character value[Note that %c in Printf]

```
char x = 'a';
printf("%c",x); // Display Result = a
```

Way 3 : Displays Next ASCII value[Note that %d in Printf]

```
char x = 'a' + 1 ;
printf("%d",x);
// Display Result = 98 ( ascii of 'b' )
```

Way 4 : Displays Next Character value[Note that %c in Printf]

```
char x = 'a' + 1;
printf("%c",x); // Display Result = 'b'
```

Reading/displaying data into/from an array

Way 5 : Displays Difference between 2 ASCII in Integer[Note %d in Printf]

```
char x = 'z' - 'a';
printf("%d",x);
/* Display Result = 25
   (difference between ASCII of z and a ) */
```

Way 6 : Displays Difference between 2 ASCII in Char [Note that %c in Printf]

```
char x = 'z' - 'a';
printf("%c",x);
/* Display Result = ↓
    ( difference between ASCII of z and a ) */
```

Reading/displaying string

Atoi Function :

- Atoi = A to I = Alphabet to Integer
- Convert String of number into Integer

Syntax:

num = atoi(String);

num - Integer Variable

String- String of Numbers

Example :

```
num = atoi("1947");
printf("%d",num);
```

Output :

1947

Significance :

1. Can Convert any String of Number into Integer Value that can Perform the arithmetic Operations like integer
2. Header File : stdlib.h

Reading/displaying string

- We cannot join two strings together by the simple arithmetic expression as follows:
- string3=string1+string2
- string2=string1+"hellow"
- Are not valid in C programming.
- The process of combining two strings together is called as concatenation. This can be done in two ways:
- 1. strcat() inbuilt function can be used
- 2. by manipulating character array indexes

- Logic to concatenate two strings : Say str1[50] and str2[50] are two input string arrays and str[100] will store concatenated string.

```
#include<stdio.h>
#include<string.h>
int main()
{
        char str1[50] = {"VISWANATH"};
    char str2[50]={"ANAND"};
        char str[100];
int i=0,j=0;
/*copy str1 into str */
for(i=0;str1[i]!='\0';i++)
                        str[i]=str1[i];
/*end str1 with a space */
str[i]= ' ';
/*copy str2 into str */
        for(j=0;str2[j]!='\0';j++)
                        str[i+j+1]=str2[j];     /*adding one to index because we have stored a space after the first name */
str[i+j+1]='\0'; /*in the last location of character array store a null value to make it a string*/
printf("\n\nThe concatenated string is : ");
printf("%s\n",str);
}
```

- Output of the program :

```
Enter first string : Lokesh is owner of
Enter second string : Ccodechamp
The concatenated string is : Lokesh is owner of Ccodechamp
```

String handling functions

Function	Work of Function
strlen()	Calculates the length of string
strcpy()	Copies a string to another string
strcat()	Concatenates(joins) two strings
strcmp()	Compares two string

Reading & Writing Strings

- ***gets() and puts()***

Functions gets() and puts() are two string functions to take string input from user and display string respectively as mentioned previously.

```
#include<stdio.h>
int main(){
   char name[30];
   printf("Enter name: ");
   gets(name);     //Function to read string from user.
   printf("Name: ");
   puts(name);     //Function to display string.
   return 0;
}
```

Though, gets() and puts() function handle string, both these functions are defined in "stdio.h" header file.

1. strlen():

In C, strlen() function calculates the length of string. It takes only one argument, i.e, string name.

Syntax of strlen()

temp_variable = strlen(string_name);

Function strlen() returns the value of type integer.

Example of strlen()

```
#include <stdio.h>
#include <string.h>
int main(){
   char a[20]="Program";
   char b[20]={'P','r','o','g','r','a','m','\0'};
   char c[20];
   printf("Enter string: ");
   gets(c);
   printf("Length of string a=%d \n",strlen(a));
   //calculates the length of string before null charcter.
   printf("Length of string b=%d \n",strlen(b));
   printf("Length of string c=%d \n",strlen(c));
   return 0;
```

Output

Enter string: String
Length of string a=7
Length of string b=7
Length of string c=6

2. strcpy()

Function strcpy() copies the content of one character array to the content of another character array. It takes two arguments as given in the syntax.

Syntax of strcpy()

```
strcpy(destination,source);
```

where destination and source are two character arrays.

Example of strcpy():

```
#include <stdio.h>
#include <string.h>
int main(){
    char a[10],b[10];
    printf("Enter string: ");
    gets(a);
    strcpy(b,a);   //Content of string a is copied to string b.
    printf("Copied string: ");
    puts(b);
    return 0;
}
```

Output

```
Enter string: Programming Tutorial
Copied string: Programming Tutorial
```

Output

Enter first string: Apple

Enter second string: Apple

Both strings are equal.

If two strings are not equal, strcmp() returns positive value if ASCII value of first mismatching element of first string is greater than that of second string and negative value if ASCII value of first mismatching element of first string is less than that of second string. For example:

```
char str1[]="and",str2[]="cat";
temp=strcmp(str1,str2);
```

Since, ASCII value of 'a' is less than that of 'c', variable temp will be negative.

4. strcat()

In C programming, strcat() concatenates(joins) two strings. It takes two arguments, i.e, two strings and resultant string is stored in the first string specified in the argument.

Syntax of strcat()

strcat(first_string,second_string);

first_string,second_string are character arrays.

Example of strcat()

```
#include <stdio.h>
#include <string.h>
int main(){
   char str1[]="This is ", str2[]="programiz.com";
   strcat(str1,str2);   //concatenates str1 and str2 and resultant string is
    stored in str1.
   puts(str1);
   puts(str2);
   return 0;
}
```

Output

```
This is programiz.com
programiz.com
```

➢ ***Other String Functions:***

The header file <string.h> contains many more string manipulation functions.

1.strncpy:

strncpy copies only the left-most n characters of the source string to the target string variable. This is a three variable function

```
strncpy(s1, s2, 5);
```

This statement copies the first 5 characters of the source string s2 to target string s1. Since the first 5 characters may not include the null string at the end , we have to place it explicitly.

```
s1[6]='\0';
```

Output

Enter string: String
Length of string a=7
Length of string b=7
Length of string c=6

2. strcpy()

Function strcpy() copies the content of one character array to the content of another character array. It takes two arguments as given in the syntax.

Syntax of strcpy()

```
strcpy(destination,source);
```

where destination and source are two character arrays.

4. strstr:

It is a two-parameter function that can be used to locate a sub-string in a string.

strstr(s1,s2); or

strstr(s1."abc");

The function strstr searches the string s1 to see whether the string s2 is contained in s1. If yes the function returns the position of the first occurrence of the sub string.

We also have functions to determine the existence of a character in a string.

strchr(s1,'m'); locates the first occurrence of character 'm' in s1.

strrchr(s1,'m'); locates the last occurrence of character 'm' in s1.

End of Chapter