
SECTION - A

Very Short Answer Questions

3 Marks (50 words each)

Q1. What is string?

Ans. String is a collection of characters. A string is a sequence of characters. Any sequence or set of characters defined within double quotation symbols is a constant string.

Strings are stored in memory as ASCII codes of characters that make up the string appended with '\0' (null character). '\0' is not same as '0'. ASCII value of null character ('\0') is zero & value of '0' is 35.

Q2. What is the difference between array and string?

Ans. Array is the collection of homogenous data elements. It means all the elements stored in array will be of same type. Array can collect multiple values of data. When we declare a variable, it will store only one value in it. But in the case of array multiple values we can store in the array.

String is a collection of characters. Array of character elements also called string.

A string is a sequence of characters. Any sequence or set of characters defined within double quotation symbols is a constant string.

Q3. How can we declare a string?

Ans. A *string variable* is always declared as an array. The variable is any valid C variable name the syntax for declaring a string variable is given below :

`char string-variable[size];`

Where the *size* is the number of characters in the *string-variable*. For example, consider the declaration of string variables given below:

```
char name[25];
```

```
char result[15];
```

The compiler provides a **null character** ('\0') at the end of the string while assigning a character string to a character array.

Q4. How can we initialize value to the strings?

Ans. It is possible to initialize the string variables while declaring. For example, consider the initialization of string variable at the time of declaration as given below:

```
char name[11] = "UNIVERSITY";
```

The variable **name** is a character array of 11 elements of storage. Therefore, the above initialization statement can be rewritten as follows:

```
char name[11] = { 'U', 'N', 'I', 'V', 'E', 'R', 'S', 'I', 'T', 'Y', '\0' };
```

While initializing a character array, by listing its elements as above, the null terminator ('\0') should be supplied explicitly.

Q5. List some standard library functions of strings?

Ans. String operations (string.h) :

C language recognizes that string is a different class of array by letting us input and output the array as

a unit and are terminated by null character. C library supports a large number of string handling functions that can be used to the string manipulations such as:

1. `strlen()` - to find length of the string (number of characters in the strings)
2. `strcat()` - concatenation of two strings.
3. `strcmp()` - comparing two strings.
4. `strcpy()` - copy (copies one string over another)

Q6. Write a short note on the `strlen()` function.

Ans. `strlen()` :

This function finds the length of a string. It counts and returns the number of characters present in a string. The length of a string refers to number of characters in the string including space. The result returned by this function is an integer value being number of character. The counting of characters start from subscript 0 and continuous upto string terminator '\0'. But null character '\0' is not a part of the character, it is merely used to mark the end of the string, so it is not counted. Thus, counting ends at the first null character.

It takes the form :

`n = strlen (string);`

where, n is an integer which receives the value of the length of the string. The argument may be a string constant.

Program to illustrate `strlen()` function :

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
void main ()
{
    char arr [ ] = "welcome";
    clrscr();
    int len1, len2;
    len1 = strlen (arr);
    len2 = strlen ("Good Morning");
    printf ("\n string = %s length = %d", arr, len1);
    printf ("\n string = %s length = %d", "Humpty
    Dumpty", len2);
    getch();
}
```

Output :

String = Welcome length = 7

String = Good Morning length = 12

Q7. Write a short note on the `strcat()` function.

Ans. `strcat()` :

It joins two strings together, i.e., it concatenates. The source string at the end of the target string. It takes two arguments, of which first is a string variable and second can be a string constant or a variable. There must be a sufficient space available to accommodate the incoming character from destination string, otherwise overflow occurs.

It takes the form :

`strcat (string 1, string 2);`

Program to illustrate `strcat()` function :

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
void main()
{
    char str1 [30] = "Wel", str2 [30] = "come";
    clrscr();
    printf ("\n first string is %s \n", str1);
    printf ("\n second string is %s \n", str2);
    strcat(str1, str2);
    printf ("first string after appending \n");
    printf ("second one : %s \n", str1);
    getch();
}
```

Output :

First string is wel

Second string is come

First string after appending

Second one : Welcome

Q8. Write a short note on the `strcmp()` function.

Ans. `strcmp()` :

It compares two strings to find out whether they are same or different. It takes two arguments of which both can be string variables or constants. The two strings are compared character by character until there is a mismatch or end of the

string is reached, whichever occurs first. If they are not, it returns the numeric difference between the ascii values of the non-matching characters. If the strings are same, it returns a value 0.

It takes the form :
strcmp (string1, string2);

Program to illustrate strcmp() function :

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
void main()
{
    char str1 [31], str2 [31];
    int value;
    clrscr();
    printf("Enter first string:");
    gets(str1);
    printf("\n Enter second string:");
    gets(str2);
    value = strcmp (str1, str2);
    if (value > 0)
    {
        printf("\n %s comes after %s", str1, str2);
        printf("\n in dictionary order \n");
    }
    else if (value < 0)
    {
        printf("\n %s comes before %s", str1, str2);
        printf("\n in dictionary order \n");
    }
    else
    {
        printf("\n both strings are same \n");
        getch();
    }
}
```

Output :

Enter first string : software

Enter first string : hardware

Hardware comes before software in dictionary order.

Q9. Write a short note on the strcpy() function.

Ans. strcpy() :

It copies the contents of one string into another. This function works almost like a string assignment operator. It takes two arguments, of which first is

a string variable and second can be a string constant or a variable. It copies the character(s) of the second argument to the first argument. The base addresses of the source and target strings should be supplied to this function.

It takes the form :
strcpy (string1, string2);

Program to illustrate strcpy() function :

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
void main()
{
    char source [ ] = "Alright";
    char target [20];
    clrscr();
    strcpy (target, source);
    printf ("\n source string = %s", source);
    printf ("\n target string = %s", target);
    getch();
}
```

Output :

Source string = Alright

Target string = Alright

Q10. What is the use of strrev() function?

Ans. The string.h header file contains function to reverse the characters in a string. The function is strrev(). This function simply reverses all the characters in the given string. The general format of strrev() function is :

strrev(string-variable);

where *string-variable* is any character array variable or string constant.

Q11. What is the difference between strcmp() & stricmp() function?

Ans. The strcmp() function takes the names of two strings to be compared as its two arguments. The comparison is made based upon the numerical value and position of each string character. When the strings are identical, the strcmp() function returns zero.

strcmp(string-variable1, string-variable2);

The *string-variable1* and *string-variable2* are any character array variables or string constants. The second string comparison function is stricmp() function. The stricmp() function is the

same as the `strcmp()` function. The only difference is that the `strcmp()` function discriminates between lowercase and uppercase letters. But, the `stricmp()` function does not discriminate between the lowercase and uppercase letters, i.e., the `stricmp()` function compares two strings without case.

If both the strings are the same, then the `stricmp()` function returns zero. Otherwise, the function returns a non-zero value.

Q12. What is the difference between `strcpy()` & `strncpy()` function?

Ans. The `strcpy()` function copies the content of one string to another. This function requires two arguments. The first argument specifies the name of the target array variable to copy the string into. The second argument specifies the name of the source array variable from where the string should be copied. The general format of `strcpy()` function is given below:

```
strcpy(string-variable1, string-variable2);
```

where *string-variable1* is the target array that may be any character array variable.

string-variable2 is the source array that may be any character array variable or string constant.

The second string copying function is `strncpy()` function. The `strncpy()` function is the same as the `strcpy()` function. The only difference is that the `strncpy()` function takes three arguments. The third argument is to specify the number of characters of the string-variable to be copied. The general format of `strncpy()` function is given below :

```
strncpy(string-variable1, string-variable2, n);
```

Here, *n* is the number of characters to be copied from *string-variable2* to *string-variable1*.

Q13. What is the difference between `strcat()` & `strncat()` function?

Ans. The `strcat()` function concatenate two strings together. This function requires two arguments. The function concatenates the `string2` with the `string1`. The general format of `strcat()` function is given below:

```
strcat(string-variable1, string-variable2);
```

where *string-variable1* and *string-variable2* are character array variables.

The `strncat()` function is the same as the `strcat()` function. The only difference is that the `strncat()` function takes three arguments. The third argument is to specify the number of characters of the string variable to be concatenated. The general format of `strncat()` function is given below :

```
Strncat(string-variable1, string-variable2, n);
```

Q14. What is the difference between a string and an array of characters?

Ans. Difference between a String and an Array of Characters :

A string is any collection of characters which is terminated by a special character null character (`'\0'`).

So string is an array of character whose last element is null character (`'\0'`).

String must be declare in a double quotes i.e. `" "`;

For Example : `Char name [] = {"Amit Kumar"};`

An Array of character is just like any other array. Here characters are store in the place of array element.

It must not terminate with null character `'\0'`.

Generally we use single quotes for Array of chars.

For Example : `Char name [] = { 'A', 'm', 'i', 't', ' ', 'k', 'u', 'm', 'a', 'r' }`

Both string and Array of character can be accessed by Index Value of Array.

Q15. List any two string handling library functions with their usage.

Ans. String Handling Library Functions :

Almost every C compiler provides a large number of library functions for manipulating strings.