

String-Handling Function:

Function Name	Action/ Operation
strcat()	concatenates (जोड़ता है) two strings
strcmp()	compare two strings
strcpy()	copies one string over another
strlen()	finds the length of a string

Q What is string?

→ In C programming, a string is a sequence of characters terminated with a null character (0).

For example,

char C[10] = "C string";

Index → 0 1 2 3 4 5 6 7

C[10] →

C	.	s	t	r	i	n	g	0
---	---	---	---	---	---	---	---	---

Q1 What are the common operations performed on character strings?

→ In C programming, a string is a sequence of characters terminated with a null character (`\0`). Strings are defined as an array of characters. Some of the most commonly used string operations are as follows:

→ `strcat()`: The `strcat()` function will append a copy of the source string to the end of destination string. The `strcat()` function takes two arguments and they are `dest` and `src`. It will append copy of the source string in the destination string.

Syntax: `strcat(dest, src);`

→ `strchr()`: `strchr` is a predefined function used for string handling. `cstring` is the header file required for string functions. This function returns a pointer to the last occurrence of character in a string.

Syntax: `strchr(const char *str, int c)`

→ strcmp() : strcmp() is a built-in library function and it is declared in <string.h> header file. and it This function takes two strings as arguments and compare these two strings.

Syntax: int strcmp(const char *leftstr, const char *rightstr);

This function takes two strings as parameters and returns an integer value based on the comparison of strings.

→ strcpy() : strcpy() is a standard library function is use to copy one string to another. In C it is present in string.h header file.

Syntax: char* strcpy(char* dest, const char* src)

→ strlen() : The strlen() function calculates the length of a given string. The strlen() function is defined in string.h header file. It doesn't count null character '\0'.

Syntax: int strlen(const char *str);

Q. Why do we need a terminating null character?

→ The null character in the C programming language is used to terminate the character strings. In other words, the Null character is used to represent the end of the string. The end of the character string or the null byte is represented by '0' or '\0' or simply NULL.

The NULL character is used for determining the length of a string. It also means that a string cannot contain a NULL. There is a NULL in memory, but it is after the last character, not in the string.

Syntax: `int strlen(const char *str);`

❏ Does C support string data type?

→ The C language does not provide (support) an inbuilt data type for strings but it has an access specifier "%s" which can be used to directly print and read strings.

example:

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


Q. What is Array? Write down the types of Arrays?

→ An array is a data structure that contains a group of elements of the same data type. Arrays are commonly used in computer programs to organize data so that a related set of values can be easily sorted and searched. Various types of arrays are as follows:

One dimensional Array: In one dimensional array the elements are stored in adjacent (संलग्न) memory locations where each element is accessed by using a single index or value. It is a linear data structure storing all the elements in sequence.

example: `int number[5];`

`int mark[] = {85, 90, 60, 93, 98};`

here, `number` and `mark` are the one-dimensional arrays.

Two dimensional Array: The two dimensional array is used for representing the elements of the array in the form of rows and columns and these are used for representing the matrix.

example: `int a[3][4];`

`int table[][3] = { {1, 2, 3}, {4, 5, 6} };`

here `a` and `table` is two dimensional array.

Multi dimensional array: C allows arrays of three or more dimensional. The exact limit is determined by the compiler. The general multi-dimensional array is

```
type array-name[s1][s2] - - [sm];  
int survey[3][5][12];
```

Here, survey is ~~the~~ three dimensional array.

Character array or string: The array of character's are called as the string. They are generally used for representing the string. String is always terminated with the null character.

Compile time initialization of one-dimensional array:

An array is a data structure that contains a group of elements of the same data type. We can initialize the elements of array in the same way as the ordinary variables when they are declared. The general form of initialization of array is:

```
data-type array-name[size] = { list of value };
```


The values in the list are separated by commas.

For example:

```
int number[5] = {1, 10, 9, 5, 2};
```

We declare number array and assign all the values. If the number of values are less than the number of sizes, then remaining elements are initialized by '0'.

```
example: int age[] = {20, 18, 19, 50};
```

```
int mark[5] = {32, 33};
```

If the number of values is greater than the number of element, then it will not work.

Run time initialization:

Array can be initialized at run time. We use `scanf()` function to initialize an array in the run time. The general form of run time initialization is

```
data_type array_name[size];  
for(int i = 0; i < size; i++){  
    scanf("Format specifier of that type", &index  
                                                of array)
```

For example:

```
int number[50];  
for(int i = 0; i < 50; i++){  
    scanf("%d", number[i]);  
}
```

If the size of array is too small then we can initialize as follows,

```
int a[3];  
scanf("%d %d %d", &a[0] &a[1] &a[2]);
```

In the first example, the int type array "mark" has size 50 and in second example array "a" has size 3. We can enter the value of these array from keyboard.

4. Initialization of two dimensional Array :

The two dimensional array is used for representing the elements of the array in the form of rows and columns and these are used for representing the matrix.

Like 1-D array, two-dimensional is used for representing by following their declaration with a list of values and enclosed in braces for example,

```
int table[2][3] = {0,0,0, 1,1,1};
```

Here, we initialized the first row to zero and the second row to one.

We can also initialize the element in the form of braces. For example,

```
int table[2][3] = { {0,0,0}, {2,2,2} };
```

We may also partially initialize a two dimension array then the remaining element are automatically set to zero.

```
int table[2][3] = { {1,1}, {2,2} };
```

Here, third row is automatically set to zero.

Q1 What do you mean by dynamic array?

→ Dynamic array are those array's which are allocated memory at the run time.

An array created at compile time by specifying size in the source code which has a fixed size and can't be modified at run time. The process of allocating memory at compile time, it is known as static memory allocation and such array are called static array. But in C, it is possible to allocate memory at run time which is known as dynamic memory allocation and the array's created at run time, called dynamic arrays.

For creating dynamic array, it used pointers, pointer variable and memory management function (malloc, calloc and realloc)

Q1 What is data structure? Explain.

→ A data structure is a specialized format for organizing, processing, retrieving and storing data. The various data structure type include following:

Array: An array stores a collection of elements in a same data type. Elements are same type which are stored together, so the position of each element can be calculated and retrieved easily by an index. Array can be fixed and flexible in length.

Stack: A stack stores a collection of items in the linear order. This order could be last in first out (LIFO).

Queue: A queue store a collection of items like a stack. However, the operation order can only be first in first out.

Linked List: A linked list stores a collection of data items in a linear order. Each element in a linked list contain a reference or link to the next item in the list.

tree: A tree stores a collection of items in abstract hierarchical way.

Heap: Heap data structure is a complete binary tree that satisfies the heap property, where any given node is always greater than its child node.

Graph: A graph stores a collection of items in a non-linear fashion. Graphs are made up of a finite set of nodes.

Tree: A tree is also known as a keyword tree and it is a data structure that stores strings as data items and that can be assigned in a visual graph.

Hash table: Hash table is also known as hash map, stores a collection of items in an associated array that plots keys to values.