Programming in C

- Arrays in C
- String
- In Built String Functions

Contents

Introduction

- Arrays a kind of data structure that can store a fixed-size sequential collection of elements of the same type.
- All arrays consist of contiguous memory locations. The lowest address corresponds to the first element and the highest address to the last element.
- Arrays are of two types:
- One-dimensional arrays
- Multidimensional arrays

Declaring an Array

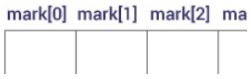
- To declare an array in C, a programmer specifies the type of the elements and the number of elements required by an array as follows –

type arrayName [arraySize];

Syntax:

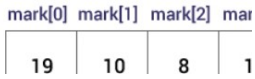
int mark[5];

char name[15];



Initialize Array

- int mark[5] = {19, 10, 8, 17, 9};



Accessing Array Elements:

- Array elements are accessed by using an integer index. Array index starts with 0 and goes till size of array minus 1. Following are few examples.

```
void main()
{
 int arr[5];
 arr[0] = 5;
 arr[2] = -10;
 arr[3/2] = 2; // this is same as arr[1] = 2
 arr[3] = arr[0];
 printf("%d %d %d %d", arr[0], arr[1], arr[2],
  arr[3]);
}
```

String

- Strings are defined as an array of characters. The difference between a character array and a string is the string is terminated with a special character '\0'.
- **Declaration of strings**: Declaring a string is as simple as declaring a one dimensional array. Below is the basic syntax for declaring a string.
- In the above **char str_name[size];**
- syntax str_name is any name given to the string variable and size is used define the length of the string, i.e the number of characters strings will store.

Initializing a String

- A string can be initialized in different ways. We will explain this with the help of an example. Below is an example to declare a string with name as str and initialize it with " welcome".

```
1. char str[] = "welcome";
2. char str[50] = "welcome";
3. char str[] = {'w','e','l','c','o','m','e','\0'};
4. char str[14]={'w','e','l','c','o','m','e','\0'};
```

Example

```
void main()
{
   // declaring string
   char str[50];
   // reading string
   scanf("%s",str);
   // print string
   printf("%s",str);
}
```

Inbuilt String Functions

- The functions present in the **string.h** header are:

Function	Use
strlen	calculates the length of string
strcat	Appends one string at the end of another
strncat	Appends first n characters of a string at the end of another
strcpy	Copies a string into another
strncpy	Copies first n characters of one string into another
strcmp	Compares two strings
strncmp	Compares first n characters of two strings
strchr	Finds the first occurrence of a given character in a string
strrchr	Finds the last occurrence of a given character in a string
strstr	Finds the first occurrence of a given string in another string

Length of String

```
#include <stdio.h>
 #include <string.h>
 void main( )
 {
char name[ ]= "Hello";
int l;
 l = strlen(name);
printf("length of %s = %d\n", name, l);
 return 0;
 }
```

Concatenation of String

```
#include <stdio.h>
 #include <string.h>
 void main( )
{
char s2[ ]= "World";
char s1[20]= "Hello";
strcat(s1, s2);
 printf("Source string = %s\n", s2);
 printf("Target string = %s\n", s1);
}
```

STRCPY

strcpy(s1, s2) copies the second string s2 to the first string s1.

```
#include <string.h>
 #include <stdio.h>
void main()
 {
 char s2[ ]= "Hello";
char s1[10];
strcpy(s1, s2);
printf("Source string = %s\n", s2);
printf("Target string = %s\n", s1);
}
```

STRCMP

strcmp(s1, s2) compares two strings and finds out whether they are same or different. It compares the two strings character by character till there is a mismatch. If the two strings are identical, it returns a 0. If not, then it returns the difference between the ASCII values of the first non-matching pair of characters.

```
 #include <stdio.h>
 #include <strings.h>
 int main()
 { char s1[ ]= "Hello";
  char s2[ ]= "World";
  int i, j;
 i = strcmp(s1, "Hello");
 j = strcmp(s1, s2);
  printf("%d \n %d\n", i, j);
 }
```

REFERENCES

1. https://www.programiz.com/article/algorithm-programming
2. https://www.tutorialspoint.com/cprogramming/c_operators.htm
3. Computer Fundamental and Programming in C, J B Dixit
4. LET US C, YASHWANT KANETKAR