Tutorial Link https://codequotient.com/tutorials/Strings and Functions/59fec760e63d6b7fd5dec143

TUTORIAL

Strings and Functions

Chapter

As strings are just a character array, we can pass strings just like other arrays to the functions and can return string from a function also. Following is an example in which a string is passed to a function and the function print the string and also return a pointer to a new string to calling function: -

```c
#include<stdio.h>

char* print_String(char str[])
{
        char str1[]="Code Quotient";
    char *b=str1;
        printf("Input String is : %s\n",str);
        return b;
}

int main()
{
        char str[] = "This is String";
        char *result;
        result=print_String(str);
        printf("String Received is : %s\n",result);
  return 0;
}

```

The above program will print the below output: -

```
Input String is : This is String
String Received is : Code Quotient
```

While using strings in functions, we have to take care that if a string is returned from a function then it must persist in memory after function termination, so that the calling function may use it. For example, the following declaration of string and returning it is OK in functions: -

```c
char* print_String()
{
  char str1[]="Code Quotient";
  char *b=str1; // stored in shared segment
  return b;
}
```

but if the function defined as below, then it might crash or segmentation fault, because the string will be destroyed on function termination and other part may use it: -

```
char* print_String()
{
        char str1[]="New String from function"; // Stored as local variable in stack segment
        return str1;
}
```