Leaning Objective:

In this Module you will be learning the following:

- Strings and its functions

Introductions:

A string in C is merely an array of characters. The length of a string is determined by a terminating null character: '\0'. So, a string with the contents, say, "abc" has four characters: 'a' , 'b' , 'c' , and the terminating null character. The terminating null character has the value zero.

Material:

Strings are always enclosed by double quotes. Whereas, character is enclosed by single quotes in C.

Declaration of strings

Before we actually work with strings, we need to declare them first. Strings are declared in a similar manner as arrays. Only difference is that, strings are of char type.

char string[20] = {'m', 's', 'i', 't', 'p', 'r', 'o', 'g', 'r', 'a', 'm', '\0'}; (or)

char string[20] = "msitprogram"; (or)

char string [] = "msitprogram";

Difference between above declarations are, when we declare char as "string[20]", 20 bytes of memory space is allocated for holding the string value.

When we declare char as "string[]", memory space will be allocated as per the requirement during execution of the program.

Example program for c string:

```
#include <stdio.h>
int main (){
        char string[20] = "fresh2refresh.com";
        printf("The string is : %s \n", string );
        return 0;
}
```

OUTPUT:

The string is: fresh2refresh.com

C STRING FUNCTIONS:

- String.h header file supports all the string functions in C language. All the string functions are given below.
- Click on each string function name below for detail description and example programs.

String functions	Description
strcat ( )	Concatenates str2 at the end of str1
strncat ( )	Appends a portion of string to another
strcpy ( )	Copies str2 into str1
strncpy( )	Copies given number of characters of one string to another
strlen( )	Gives the length of str1

strcmp( )	Returns 0 if str1 is same as str2. Returns <0 if strl < str2. Returns >0 if str1 > str2
strcmpi( )	Same as strcmp() function. But, this function negotiates case. "A" and "a" are treated as same.
strchr( )	Returns pointer to first occurrence of char in str1
strrchr( )	last occurrence of given character in a string is found
strstr( )	Returns pointer to first occurrence of str2 in str1
strrstr( )	Returns pointer to last occurrence of str2 in str1
strdup( )	Duplicates the string
strlwr( )	Converts string to lowercase
strupr( )	Converts string to uppercase
strrev( )	Reverses the given string
strset( )	Sets all character in a string to given character
strnset( )	It sets the portion of characters in a string to given character
strtok( )	Tokenizing given string using delimiter

C – strcat() function

- strcat( ) function in C language concatenates two given strings. It concatenates source string at the end of destination string. Syntax for strcat( ) function is given below. char * strcat (char * destination, const char * source);
- Example:
  strcat (str2, str1); – str1 is concatenated at the end of str2. strcat (str1, str2); – str2 is concatenated at the end of str1.
- As you know, each string in C is ended up with null character ('\0').
- In strcat( ) operation, null character of destination string is overwritten by source string's first character and null character is added at the end of new destination string which is created after strcat( ) operation.

C – strncat() function

- strncat( ) function in C language concatenates (appends) portion of one string at the end of another string. Syntax for strncat( ) function is given below. char * strncat (char * destination, const char * source, size_t num);
- Example:
  strncat (str2, str1, 3); – First 3 characters of str1 is concatenated at the end of str2. strncat (str1, str2, 3); – First 3 characters of str2 is concatenated at the end of str1.
- As you know, each string in C is ended up with null character ('\0').
- In strncat( ) operation, null character of destination string is overwritten by source string's first character and null character is added at the end of new destination string which is created after strncat( ) operation.

C – strcpy() function

- strcpy( ) function copies contents of one string into another string. Syntax for strcpy function is given below.

char * strcpy (char * destination, const char * source);

- Example:
  strcpy (str1, str2) – It copies contents of str2 into str1.
  strcpy (str2, str1) – It copies contents of str1 into str2.
- If destination string length is less than source string, entire source string value won't be copied into destination string.
- For example, consider destination string length is 20 and source string length is 30. Then, only 20 characters from source string will be copied into destination string and remaining 10 characters won't be copied and will be truncated.

C – strncpy() function

- strncpy( ) function copies portion of contents of one string into another string. Syntax for strncpy( ) function is given below.

  char * strncpy (char * destination, const char * source, size_t num);

- Example:
  strncpy (str1, str2, 4) – It copies first 4 characters of str2 into str1.
  strncpy (str2, str1, 4) – It copies first 4 characters of str1 into str2.
- If destination string length is less than source string, entire source string value won't be copied into destination string.
- For example, consider destination string length is 20 and source string length is 30. If you want to copy 25 characters from source string using strncpy( ) function, only 20 characters from source string will be copied into destination string and remaining 5 characters won't be copied and will be truncated.

C – strlen() function

- strlen( ) function in C gives the length of the given string. Syntax for strlen( ) function is given below.

  size_t strlen (const char * str);

- strlen( ) function counts the number of characters in a given string and returns the integer value.
- It stops counting the character when null character is found. Because, null character indicates the end of the string in C.

C – strcmp() function

- strcmp( ) function in C compares two given strings and returns zero if they are same.
- If length of string1 < string2, it returns < 0 value. If length of string1 > string2, it returns > 0 value. Syntax for strcmp( ) function is given below.
  int strcmp (const char * str1, const char * str2);
- strcmp( ) function is case sensitive. i.e, "A" and "a" are treated as different characters.

C – strrstr() function

- strrstr( ) function returns pointer to the last occurrence of the string in a given string. Syntax for strrstr( ) function is given below.

  char *strrstr(const char *str1, const char *str2);

- strrstr( ) function is non-standard function which may not available in standard library in C.

C – strdup() function

- strdup( ) function in C duplicates the given string. Syntax for strdup( ) function is given below.
  char *strdup(const char *string);

- strdup( ) function is non-standard function which may not available in standard library in C.

C – strlwr() function

- strlwr( ) function converts a given string into lowercase. Syntax for strlwr( ) function is given below.

  char *strlwr(char *string);

- strlwr( ) function is non-standard function which may not available in standard library in C.

C – strupr() function

- strupr( ) function converts a given string into uppercase. Syntax for strupr( ) function is given below.

  char *strupr(char *string);

- strupr( ) function is non-standard function which may not available in standard library in C.

C – strrev() function

- strrev( ) function reverses a given string in C language. Syntax for strrev( ) function is given below.

  char *strrev(char *string);

- strrev( ) function is non-standard function which may not available in standard library in C.

C – strset() function

- strset( ) function sets all the characters in a string to given character. Syntax for strset( ) function is given below.

  char *strset(char *string, int c);

- strset( ) function is non-standard function which may not available in standard library in C.

C – strnset() function

- strnset( ) function sets portion of characters in a string to given character. Syntax for strnset( ) function is given below.

  char *strnset(char *string, int c);

- strnset( ) function is non-standard function which may not available in standard library in C.

Example program for strncpy() function in c:

```
#include <stdio.h>
#include <string.h>
void main( ){
  char source[ ] = "fresh2refresh" ;
  char target[20]= "" ;
  printf ( "\nsource string = %s", source ) ;
  printf ( "\ntarget string = %s", target ) ;
  strncpy ( target, source, 5 ) ;
  printf ( "\ntarget string after strcpy( ) = %s", target ) ;
}
```

OUTPUT:

```
source string = fresh2refresh
target string =
target string after strncpy( ) = fresh
```

Problem Set:

1. Write a C program to find length of a string.
2. Write a C program to copy one string to another string.
3. Write a C program to concatenate two strings.
4. Write a C program to compare two strings.
5. Write a C program to convert lowercase string to uppercase.
6. Write a C program to find total number of alphabets, digits or special character in a string.
7. Write a C program to count total number of vowels and consonants in a string.
8. Write a C program to count total number of words in a string.
9. Write a C program to find reverse of a string.
10. Write a C program to check whether a string is palindrome or not.
11. Write a C program to reverse order of words in a given string.
12. Write a C program to find first occurrence of a character in a given string.
13. Write a C program to count occurrences of a character in given string.
14. Write a C program to count frequency of each character in a string.
15. Write a C program to remove first occurrence of a character from string.
16. Write a C program to remove all occurrences of a character from string.
17. Write a C program to remove all repeated characters from a given string.
18. Write a C program to replace first occurrence of a character with another in a string.
19. Write a C program to replace all occurrences of a character with another in a string.
20. Write a C program to find first occurrence of a word in a given string.
21. Write a C program to count occurrences of a word in a given string.
22. Write a C program to remove all occurrence of a word in given string.
23. Write a C program to trim both leading and trailing white space characters in a string.