3.6.STRING HANDLING FUNCTIONS IN C

C programming language provides a set of pre-defined functions called string handling functions to work with string values.

The string handling functions are defined in a header file called string.h. Whenever we want to use any string handling function we must include the header file called string.h.

The following table provides most commonly used string handling function and their use...

Function	Syntax (or) Example	Description
strc()	strcpy(string1, string2)	Copies string2 value into string1
strncpy()	strncpy(string1, string2, 5)	Copies first 5 characters string2 into string1
strlen()	strlen(string1)	returns total number of characters in string1
strcat()	strcat(string1,string2)	Appends string2 to string1
strncat()	strncpy(string1, string2, 4)	Appends first 4 characters of string2 to string1
strcmp()	strcmp(string1, string2)	Returns 0 if string1 and string2 are the same; less than 0 if string1<string2; greater than 0 if string1>string2
strncmp()	strncmp(string1, string2, 4)	Compares first 4 characters of both string1 and string2
strcmpi()	strcmpi(string1,string2)	Compares two strings, string1 and string2 by ignoring case (upper or lower)
stricmp()	stricmp(string1, string2)	Compares two strings, string1 and string2 by ignoring case (similar to strcmpi())
strlwr()	strlwr(string1)	Converts all the characters of string1 to lower case.
strupr()	strupr(string1)	Converts all the characters of string1 to upper case.
strdup()	string1 = strdup(string2)	Duplicated value of string2 is assigned to string1

strchr()	strchr(string1, 'b')	Returns a pointer to the first occurrence of character 'b' in string1
strrchr()	'strrchr(string1, 'b')	Returns a pointer to the last occurrence of character 'b' in string1
strstr()	strstr(string1, string2)	Returns a pointer to the first occurrence of string2 in string1

Function	Syntax (or) Example	Description
strset()	strset(string1, 'B')	Sets all the characters of string1 to given character 'B'.
strnset()	strnset(string1, 'B', 5)	Sets first 5 characters of string1 to given character 'B'.
strrev()	strrev(string1)	It reverses the value of string1

1.<u>String Length</u>

For example, to get the length of a string, you can use the strlen() function:

Example

```
#include <stdio.h>
#include <string.h>
int main() {
  char alphabet[] = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
  printf("%d", strlen(alphabet));
  return 0;
}
```

Output

26

we used sizeof to get the size of a string/array. Note that sizeof and strlen behaves differently, as sizeof also includes the \0 character when counting:

```
#include <stdio.h>
#include <string.h>
int main() {
```

```
  char alphabet[] = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
  printf("Length is: %d\n", strlen(alphabet));
  printf("Size is: %d\n", sizeof(alphabet));
  return 0;
}
```

Output

LENETH IS:26

SIZE IS:27

2. Concatenate Strings

To concatenate (combine) two strings, you can use the strcat() function:

```
#include <stdio.h>
#include <string.h>
int main() {
  char str1[20] = "Hello ";
  char str2[] = "World!";
  // Concatenate str2 to str1 (the result is stored in str1)
  strcat(str1, str2);
  printf("%s", str1);
  return 0;
}
```

Output

Hello world!

3. Copy Strings

To copy the value of one string to another, you can use the strcpy() function:

```
#include <stdio.h>
#include <string.h>
int main() {
  char str1[20] = "Hello World!";
  char str2[20];
  strcpy(str2, str1);
  printf("%s", str2);
  return 0;
```

```
}
```

Output

Hello world!

Compare Strings

To compare two strings, you can use the strcmp() function.

It returns 0 if the two strings are equal, otherwise a value that is not 0:

```
#include <stdio.h>
#include <string.h>
int main() {
  char str1[] = "Hello";
  char str2[] = "Hello";
  char str3[] = "Hi";
  printf("%d\n", strcmp(str1, str2)); // Returns 0 (the strings are equal)
 printf("%d\n", strcmp(str1, str3)); // Returns -1 (the strings are not equal)
 return 0;
}
```

4.First Character Occurrence: strchr()

This function locates the first occurrence of a character in a string.

Example:

```
#include <stdio.h>
#include <string.h>
int main() {
    char sentence[] = "This is a sample sentence.";
    char *ptr = strchr(sentence, 'a');
    printf("strchr(): %s\n", ptr);
```

```
    return 0;
}
```

Output

strchr(): a sample sentence.

5. Last Character Occurrence: strrchr()

This function locates the last occurrence of a character in a string.

Example:

```
#include <stdio.h>
#include <string.h>
int main() {
    char sentence[] = "This is a sample sentence.";
    char *lastPtr = strrchr(sentence, 'a');
    printf("strrchr(): %s\n", lastPtr);
    return 0;
}
```

Output

strrchr(): ample sentence.

6. String Search: strstr()

This function searches for the first occurrence of a substring within a string.

Example:

```
#include <stdio.h>
#include <string.h>
int main() {
    char text[] = "The quick brown fox jumps over the lazy dog.";
    char *substr = strstr(text, "fox");
```

```
    printf("strstr(): %s\n", substr);
    return 0;
}
```

Output:

```
Output
strstr(): fox jumps over the lazy dog.
```


7. String Token Break: strtok()

This function breaks a string into a series of tokens based on a delimiter.

Example:

```
#include <stdio.h>
#include <string.h>
int main() {
    char sentence[] = "This is a sample sentence";
    char *token = strtok(sentence, " ");
    while (token != NULL) {
        printf("strtok(): %s\n", token);
        token = strtok(NULL, " ");
    }
    return 0;
}
```

Output:

```
strtok(): This
strtok(): is
strtok(): a
strtok(): sample
strtok(): sentence
```

8. Lowercase String: strlwr()

This function converts a string to lowercase. Note, this function is not included in the Standard Library.

Example:

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>

void toLowerCase(char *str) {
    for (int i = 0; str[i]; i++) {
        str[i] = tolower((unsigned char)str[i]);
    }
}

int main() {
    char str5[] = "LoWeRcAsE";
    toLowerCase(str5);
    printf("strlwr(): %s\n", str5);
    return 0;
}
```

Output:

```
strlwr(): lowercase
```

9. Uppercase String: strupr()

This function converts a string to uppercase. Note, this function is not included in the Standard Library.

Example:

```
#include <stdio.h>
#include <string.h>
```

```
#include <ctype.h>
void toUpperCase(char *str) {
    for (int i = 0; str[i]; i++) {
        str[i] = toupper((unsigned char)str[i]);
    }
}
int main() {
    char str6[] = "UpperCase";
    toUpperCase(str6);
    printf("strupr(): %s\n", str6);
    return 0;
}
```

Output:

strupr(): UPPERCASE

10. <u>Duplicate String: strdup()</u>

This function duplicates a string.

Example:

```
#include <stdio.h>
#include <string.h>
int main() {
    char original[] = "Original String";
    char *duplicate = strdup(original);
    printf("strdup(): %s\n", duplicate);
    return 0;
}
```

Output:

strdup(): Original String