



SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai



DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

COURSE NAME : 23ITT101- PROBLEM SOLVING & C PROGRAMMING

I YEAR /I SEMESTER

Unit II – ARRAYS AND STRINGS

Topic : String Operations



String operations

- String operations refer to the set of functions that allow you to manipulate and work with strings (which are arrays of characters).
- The C Standard Library provides several functions for common string operations such as copying, concatenation, comparison, and searching.



The C standard library (<string.h>)

Functions for working with strings:

- **strlen():** To get the length of a string (excluding the null terminator).

```
int len = strlen(str);  
printf("Length of string: %d", len);
```
- **strcpy():** To copy one string to another.

```
char src[] = "Hello";  
char dest[10];  
strcpy(dest, src);  
printf("%s", dest); // Output: Hello
```



The C standard library (<string.h>)

Functions for working with strings:

- **strcat():** To concatenate (append) one string to another.

```
char str1[20] = "Hello";  
char str2[] = " World";  
strcat(str1, str2);  
printf("%s", str1); // Output: Hello World
```
- **strcmp():** To compare two strings.

```
char str1[] = "apple";  
char str2[] = "banana";  
int result = strcmp(str1, str2);  
// Returns a negative value since "apple" < "banana"
```



The C standard library (<string.h>)

Functions for working with strings:

- **strchr():** To find the first occurrence of a character in a string.

```
char str[] = "Hello";  
char *ptr = strchr(str, 'e');  
if (ptr != NULL) {  
    printf("Found 'e' at position: %ld", ptr - str);  
}
```
- **strncpy() :** Copy a String with LimitPurpose: Copies a specified number of characters from the source string to the destination string.



The C standard library (<string.h>)



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

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

```
int main()
```

```
{
```

```
    char src[] = "Hello, world!";
```

```
    char dest[6];
```

```
    strncpy(dest, src, 5); // Copy first 5 characters from src to dest
```

```
    dest[5] = '\0'; // Null terminate the destination string
```

```
    printf("Destination string: %s\n", dest);
```

```
    return 0;
```

```
}
```

Output:

Destination string: Hello



The C standard library (<string.h>)

Functions for working with strings:

- **strncat()** - Concatenate a Limited Number of Characters

Purpose: Appends a specified number of characters from the source string to the destination string.

Example:

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

```
int main() {
    char str1[20] = "Hello";
    char str2[] = " World!";

    strncat(str1, str2, 3); // Concatenate first 3 characters of str2 to str1
    printf("Concatenated string: %s\n", str1);
    return 0;
}
```



The C standard library (<string.h>)



Functions for working with strings:

- **strncmp()** - Compare a Limited Number of Characters

Purpose: Compares the first n characters of two strings.

Example:

```
#include <stdio.h>
#include <string.h>
int main() {
    char str1[] = "Hello";
    char str2[] = "Hel";

    int result = strncmp(str1, str2, 3); // Compare first 3 characters
    if (result == 0) {
        printf("The strings are equal.\n");
    } else if (result < 0) {
        printf("The first string is smaller.\n");
    } else {
        printf("The first string is larger.\n");
    }
    return 0;
}
```

Output:

The strings are equal.



The C standard library (<string.h>)



Functions for working with strings:

- **strstr()** - Find a Substring in a String

Purpose: Finds the first occurrence of a substring in a string.

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

int main() {
    char str[] = "Hello, world!";
    char *result = strstr(str, "world"); // Find substring "world"

    if (result != NULL) {
        printf("Substring found at position: %ld\n", result - str);
    } else {
        printf("Substring not found.\n");
    }

    return 0;
}
```

Output:

Substring found at position: 7



The C standard library (<string.h>)



Functions for working with strings:

- **strtok()** - Tokenize a StringPurpose: Breaks a string into a series of tokens (substrings) based on delimiters.

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

int main() {
    char str[] = "Hello, world! How are you?";
    char *token = strtok(str, ",!?!");
    // Split by space, comma, exclamation mark, or question mark

    while (token != NULL) {
        printf("Token: %s\n", token);
        token = strtok(NULL, ",!?!");
    }

    return 0;
}
```

Output:

```
Token: Hello
Token: world
Token: How
Token: are
Token: you
```



Summary



- C language supports a large number of string handling functions that can be used to carry out many of the string manipulations.
- **Character handling functions are included in the library `<ctype.h>`**
- These functions are packaged in **`<string.h>`** library.

Function	Purpose
strlen()	Gets the length of a string.
strcpy()	Copies one string to another.
strncpy()	Copies a limited number of characters from one string to another.
strcat()	Concatenates two strings.
strncat()	Concatenates a limited number of characters from one string to another.
strcmp()	Compares two strings lexicographically.
strncmp()	Compares the first n characters of two strings.
strchr()	Finds the first occurrence of a character in a string.
strstr()	Finds the first occurrence of a substring in a string.
strtok()	Breaks a string into tokens based on delimiters.



SNSCE/ AI&DS/ AP / Dr . N. ABIRAMI