



Birla Institute of Technology & Science, Pilani
Hyderabad Campus

snprintf()

innovate

achieve

lead

What is snprintf() in C?



In C, **snprintf()** function is a standard library function that is used to print the specified string till a specified length in the specified format. It is defined in the **<stdio.h>** header file.

Syntax

```
snprintf(str, size, format, ...);
```

Parameters

str: It is a pointer to the buffer.

size: It is the maximum number of bytes (characters) that will be written to the buffer.

format: C string that contains a format string that follows the same specifications as format in printf.

(...): The optional (...) arguments are just the string formats like ("%d", myint) as seen in printf.

Return Value

The number of characters that would have been written on the buffer, if 'n' had been sufficiently large.

If an encoding error occurs, a negative number is returned.

Characteristics of snprintf()



- The snprintf() function formats and stores a series of characters and values in the array buffer.
- The snprintf() function accepts an argument 'n', which indicates the maximum number of characters (including at the end of null character) to be written to buffer.
- The snprintf() function is used to redirect the output of printf() function onto a buffer.
- The snprintf() also returns the number characters that were supposed to be written onto the buffer (excluding the null terminator), irrespective of the value of 'n' passed.
- Only when the returned value is non-negative and less than 'n', the string has been completely written as expected.

Example1 on snprintf()

```
// C program to demonstrate snprintf()
#include <stdio.h>

int main()
{
    char buffer[50];
    char* s = "geeksforgeeks";

    // Counting the character and storing
    // in buffer using snprintf
    printf("Writing %s onto buffer"
           " with capacity 6",
           s);
    int j = snprintf(buffer, 6, "%s\n", s);

    // Print the string stored in buffer and
    // character count
    printf("\nString written on "
           "buffer = %s",
           buffer);
    printf("\nValue returned by "
           "snprintf() method = %d\n",
           j);

    return 0;
}
```

Output

Writing geeksforgeeks onto buffer with capacity 6

String written on buffer = geeks

Value returned by snprintf() method = 14

Example2 on snprintf()



```
// C program to demonstrate snprintf()
#include <stdio.h>

int main()
{
    char buffer[50];

    // join two or more strings
    char* str1 = "quick";
    char* str2 = "brown";
    char* str3 = "lazy";
    int max_len = sizeof buffer;

    int j = snprintf(buffer, max_len,
        "The %s %s fox jumped over the %s dog.",
        str1, str2, str3);
    printf("\nThe number of bytes printed to 'buffer' "
        "(excluding the null terminator) is %d\n",
        j);
    if (j >= max_len)
        fputs("Buffer length exceeded; string truncated",
            stderr);
    puts("Joined string:");
    puts(buffer);

    return 0;
}
```

Output

The number of bytes printed to 'buffer' (excluding the null terminator) is 45
Joined string:
The quick brown fox jumped over the lazy dog.