

Description

The C library function **int scanf(const char *format, ...)** reads formatted input from stdin.

Declaration

Following is the declaration for scanf() function.

```
int scanf(const char *format, ...)
```

Parameters

format -- This is the C string that contains one or more of the following items: *Whitespace character, Non-whitespace character and Format specifiers*. A format specifier will be like **[=%[*][width][modifiers]type=]** as explained below:

argument	Description
*	This is an optional starting asterisk indicates that the data is to be read from the stream but ignored, i.e. it is not stored in the corresponding argument.
width	This specifies the maximum number of characters to be read in the current reading operation
modifiers	Specifies a size different from int (in the case of d, i and n), unsigned int (in the case of o, u and x) or float (in the case of e, f and g) for the data pointed by the corresponding additional argument: h : short int (for d, i and n), or unsigned short int (for o, u and x) l : long int (for d, i and n), or unsigned long int (for o, u and x), or double (for e, f and g) L : long double (for e, f and g)
type	A character specifying the type of data to be read and how it is expected to be read. See next table.

scanf type specifiers:

type	Qualifying Input	Type of argument
c	Single character: Reads the next character. If a width different from 1 is specified, the function reads width characters and stores them in the successive locations of the array passed as argument. No null character is appended at the end.	char *

d	Decimal integer: Number optionally preceeded with a + or - sign	int *
e,E,f,g,G	Floating point: Decimal number containing a decimal point, optionally preceeded by a + or - sign and optionally folowed by the e or E character and a decimal number. Two examples of valid entries are -732.103 and 7.12e4	float *
o	OctalInteger:	int *
s	String of characters. This will read subsequent characters until a whitespace is found (whitespace characters are considered to be blank, newline and tab).	char *
u	Unsigned decimal integer.	unsign ed int *
x,X	Hexadecimal Integer	int *

additional arguments -- Depending on the format string, the function may expect a sequence of additional arguments, each containing one value to be inserted instead of each %-tag specified in the format parameter, if any. There should be the same number of these arguments as the number of %-tags that expect a value.

Return Value

If successful, the total number of characters written is returned, otherwise a negative number is returned.

Example

The following example shows the usage of scanf() function.

```
#include <stdio.h>

int main()
{
    char str1[20], str2[30];

    printf("Enter name: ");
    scanf("%s", &str1);

    printf("Enter your website name: ");
    scanf("%s", &str2);

    printf("Entered Name: %s\n", str1);
    printf("Entered Website:%s", str2);
}
```

```
    return(0);  
}
```