

Function scanf () in C

The function `int scanf (const char * format, ...)` in Library C standard reads input that has been formatted from stdin.

The function **int scanf (const char * format, .)** in Library C standard reads input that has been formatted from stdin.

The function **scanf does not accept the space** between two strings (different from the function gets ()), ie you can only enter one consecutive string, if you enter the whole space, the content after the first blank will not accept.

Declare the function scanf () in C

Below is the declaration for the scanf () function in C:

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

Parameters

format - This is the string containing one of the following items:

Character Whitespace, **Non-whitespace** and **Format specifier** . A format specifier will be the form [= % [*] [width] [modifiers] type =] , explained as follows:

Parameters Description * This is a starting asterisk (optional) to indicate that the data is read from the Stream but ignored, for example, it is not stored in the corresponding parameter width Specifies the number of characters maximum read in current readings modifiers Define a different size from int (in the case of d, I and n), unsigned int (in the case of o, u and x) or float (in case e, f and g) for data pointed by the corresponding additional parameter: h for int short (for d, i and n), or unsigned short int (for o, u and x); l for int long (for d, i and n), or unsigned long int (for o, u and x), or double (for e, f and g); L for 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. You follow the next table Type specifier for fscanf function

Type Input Shock type Single character: Read the next character. If a width other than 1 is specified, this function reads the width of characters and stores them in consecutive array positions that have been passed as parameters. No null characters are appended to the end of char * d Decimal integer numbers: This number is arbitrarily preceded by a + sign or - int * e, E, f, g, G floating point numbers: The decimal number contains a decimal point, arbitrarily preceded by a + or - sign and followed arbitrarily by the letter e or E and a decimal number. Two valid examples are -732.103 and 7.12e4 float * o Octal integer int * s Character string. It will read consecutive characters until a whitespace is found (maybe blank, newline) and tab) char * u Decimal integers unsigned int * x, X Hexadecimal integers segment int *

Additional parameters - Depending on the format string, this function may have an additional parameter array, each containing a value to be inserted instead of each % -tag specified in the format parameter. , if. This number of parameters should be the same number as % -tags that expect a value.

Returns the value

If successful, the total number of characters recorded will be returned, if it fails, returns a negative number.

For example

In the following example, I have entered spaces in the phone number and date of birth, and you follow the results display to see that the **scanf** command does not accept spaces.

The following C program illustrates the usage of the scanf () function in C:

```
#include <stdio.h>
int main () { char str1 [ 20 ], str2 [ 30 ]; printf ( "Nhap ten: "
```

Compiling and running the above C program will result:

```
Nhap ten: ThienNgocMinhUy
Nhap so dien thoai va ngay sinh: 1900100thay 01/01/100thay
Ten vua nhap: ThienNgocMinhUy
So dien thoai va ngay sinh vua nhap: 1900100thay
-----
```

According to Tutorialspoint

Previous lesson: Function fscanf () in C

Next lesson: The function sscanf () in C

You finished reading the article "**Function scanf () in C**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.