**一.printf/wprintf函数简介**

[C 语言](https://www.codersrc.com/c%e8%af%ad%e8%a8%80%e5%9f%ba%e7%a1%80)对于输出常规字符串到控制台，大家都知道可以通过 [printf 函数](https://www.codersrc.com/archives/7464.html)，那么如果需要输出 [unicode 字符串](https://www.codersrc.com/archives/10665.html)到控制台如何完成呢？？

wchar\_t\* p = L"123456";

printf(L"unicode :%s \n", p);

/\*

编译器报错：

error C2664: “int printf(const char \*,...)”: 无法将参数 1 从“const wchar\_t [14]”转换为“const char \*”

\*/

在 [《 Unicode 和多字节区别 》](https://www.codersrc.com/archives/10665.html) 文章中有介绍 ：**Win32 API中大部分参数有字符串的函数都有两个版本，以 A 结尾,代表多字节版本，以 W 结尾,代表 Unicode 版本,两者互不兼容**；[wprintf](https://www.codersrc.com/archives/10701.html) 函数声明如下：

int wprintf( const wchar\_t \*format, ... );

***wprintf 使用方式和 printf 一样，如果输出多字节字符串 char ，使用 printf ；如果输出宽字节字符串 wchar\_t，使用 wprintf ；***

**二.printf/wprintf函数输出字符串**

* **输出多字节字符串 char ，使用 printf 函数 ；**
* **输出宽字节字符串 wchar\_t，使用 wprintf 函数；**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//@Author:猿说编程

//@Blog(个人博客地址): www.codersrc.com

//@File:C语言教程 - C/C++ printf / wprintf 函数使用

//@Time:2021/08/05 08:00

//@Motto:不积跬步无以至千里，不积小流无以成江海，程序人生的精彩需要坚持不懈地积累！

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include "stdafx.h"

#include <stdio.h>

#include "windows.h"

int main()

{

char\* p1 = "123456"; //普通字符串 - 多字节字符串

wchar\_t\* p2 = L"123456"; //unicode字符串 - 字符串有修饰符 L

printf("p1 :%s \n", p1);

wprintf(L"p2 :%s \n", p2);

system("pause");

return 0;

}

/\*

输出：

p1 :123456

p2 :123456

请按任意键继续. . .

\*/

**三.printf/wprintf函数输出中文乱码**

到目前为止不管是输出多字节字符串还是 [unicode 字符串](https://www.codersrc.com/archives/10665.html)都能实现，那么在试一试中文看看是否正常？直接将上面的代码略作修改：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//@Author:猿说编程

//@Blog(个人博客地址): www.codersrc.com

//@File:C语言教程 - C/C++ printf / wprintf 函数使用

//@Time:2021/08/05 08:00

//@Motto:不积跬步无以至千里，不积小流无以成江海，程序人生的精彩需要坚持不懈地积累！

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

char\* p1 = "C/C++教程-猿说编程"; //普通字符串 - 多字节字符串

wchar\_t\* p2 = L"C/C++教程-猿说编程"; //unicode字符串 - 字符串有修饰符 L

printf("p1 :%s \n", p1);

wprintf(L"p2 :%s \n", p2);

/\*

输出：

p1 :C/C++教程-猿说编程

p2 :C/C++??-????

请按任意键继续. . .

\*/

***解决wprintf输出中文乱码：***

***如果***[***Unicode***](https://www.codersrc.com/archives/10665.html)***字符串中含有中文，使用 wprintf 函数输出前通常要通过***[***setlocale***](https://www.codersrc.com/archives/10701.html)***函数设置输出文字的语言区域，****例如：中文*

#include <Locale.h> //需要包含头文件

//以下两种方式都可以

\_wsetlocale(LC\_ALL, L"chs"); //chs 表示中文 方法一

setlocale(LC\_ALL, "chs"); //chs 表示中文 方法二

完整代码如下：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//@Author:猿说编程

//@Blog(个人博客地址): www.codersrc.com

//@File:C语言教程 - C/C++ printf / wprintf 函数使用

//@Time:2021/08/05 08:00

//@Motto:不积跬步无以至千里，不积小流无以成江海，程序人生的精彩需要坚持不懈地积累！

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include "stdafx.h"

#include <stdio.h>

#include "windows.h"

#include <Locale.h>

int main()

{

setlocale(LC\_ALL, "chs"); // 本地化语言设置为中文，否则中文乱码

char\* p1 = "C/C++教程-猿说编程"; //普通字符串 - 多字节字符串

wchar\_t\* p2 = L"C/C++教程-猿说编程"; //unicode字符串 - 字符串有修饰符 L

printf("p1 :%s \n", p1);

wprintf(L"p2 :%s \n", p2);

system("pause");

return 0;

}

/\*

输出：

p1 :C/C++教程-猿说编程

p2 :C/C++教程-猿说编程

请按任意键继续. . .

\*/

**四.猜你喜欢**

1. [C语言 数组下标越界和内存溢出区别](https://www.codersrc.com/archives/8331.html)
2. [C语言 使用指针遍历数组](https://www.codersrc.com/archives/8540.html)
3. [C语言 指针和数组区别](https://www.codersrc.com/archives/8564.html)
4. [C语言 指针数组和数组指针区别](https://www.codersrc.com/archives/8613.html)
5. [C语言 野指针](https://www.codersrc.com/archives/8700.html)
6. [C语言 函数值传递和址传递](https://www.codersrc.com/archives/8738.html)
7. [C语言 函数不定长参数](https://www.codersrc.com/archives/8780.html)
8. [C语言 函数指针](https://www.codersrc.com/archives/8793.html)
9. [C语言 指针函数](https://www.codersrc.com/archives/8809.html)
10. [C语言 回调函数 callback](https://www.codersrc.com/archives/8832.html)
11. [C语言 #pragma once](https://www.codersrc.com/archives/8931.html)
12. [C语言 #include <> 与 #include “” 区别](https://www.codersrc.com/archives/9045.html)
13. [C语言 const 修饰函数参数](https://www.codersrc.com/archives/9106.html)
14. [C语言 const 和 define 区别](https://www.codersrc.com/archives/9123.html)
15. [C语言 #运算符](https://www.codersrc.com/archives/9423.html)
16. [C语言 ##运算符](https://www.codersrc.com/archives/9438.html)
17. [C语言 \_\_VA\_ARGS\_\_](https://www.codersrc.com/archives/9450.html)
18. [C语言 ##\_\_VA\_ARGS\_\_](https://www.codersrc.com/archives/9480.html)
19. [C语言 函数不定长参数 ##\_\_VA\_ARGS\_\_经典案例](https://www.codersrc.com/archives/9499.html)
20. [C语言 va\_start / va\_end / va\_arg 自定义 printf 函数](https://www.codersrc.com/archives/9575.html)
21. [C语言 main 函数](https://www.codersrc.com/archives/9680.html)
22. [C语言 main 函数参数 main(int argc, char \*argv[])](https://www.codersrc.com/archives/9687.html)
23. [C语言 局部变量](https://www.codersrc.com/archives/9815.html)
24. [C语言 全局变量](https://www.codersrc.com/archives/9836.html)
25. [C语言 全局变量和局部变量区别](https://www.codersrc.com/archives/9856.html)
26. [C语言 static](https://www.codersrc.com/archives/9895.html)
27. [C语言 extern](https://www.codersrc.com/archives/9944.html)
28. [C/C++ Unicode 和多字节区别](https://www.codersrc.com/archives/10665.html)
29. [C/C++ wprintf 输出中文乱码](https://www.codersrc.com/archives/10701.html)
30. [C/C++ printf / wprintf 函数使用](https://www.codersrc.com/archives/10704.html)