## int printf(const char \*format [, argument]... );

链接： [printf](http://msdn.microsoft.com/zh-cn/library/wc7014hz(v=vs.110).aspx)

头文件：<stdio.h>

**功能：设置格式并打印一系列字符和值到标准输出流，stdout。 如果参数后面 *format* 字符串，*format* 字符串必须包含确保参数的输出格式的规范。 printf 和 [fprintf](http://msdn.microsoft.com/zh-cn/library/xkh07fe2(v=vs.110).aspx) 具有相同的行为，但 printf 写入输出与 stdout 而不是类型 FILE的目标。**

**返回：**返回打印的字符数，如果发生错误则返回负值，遇到EOF返回-1。

format 参数输出的格式，定义格式为：

**%[**[flags](http://msdn.microsoft.com/zh-cn/library/8aky45ct.aspx)**] [**[width](http://msdn.microsoft.com/zh-cn/library/25366k66.aspx)**] [.**[precision](http://msdn.microsoft.com/zh-cn/library/0ecbz014.aspx)**] [{h | l | ll | w | I | I32 | I64}]**[type](http://msdn.microsoft.com/zh-cn/library/hf4y5e3w.aspx)

## int scanf( const char \*format [,argument]... );

链接： [scanf](http://msdn.microsoft.com/zh-cn/library/9y6s16x1.aspx)

头文件：<stdio.h>

**功能：从标准输入流中读取格式化数据**

**返回：**返回成功赋值的数据项数，出错或遇到end of file则返回EOF，返回值为 0 表示未分配字段

format 参数输出的格式，定义格式为：

**% [\*] [] [{**[**h|l|ll|I64|L**](http://msdn.microsoft.com/zh-cn/library/xdb9w69d.aspx)**}]**[**类型**](http://msdn.microsoft.com/zh-cn/library/6ttkkkhh.aspx)[**宽度**](http://msdn.microsoft.com/zh-cn/library/xdb9w69d.aspx)

格式字符说明：

|  |  |
| --- | --- |
| %a,%A | 读入一个浮点值(仅C99有效) |
| %i | 读入十进制，八进制，十六进制整数 |
| %o | 读入八进制整数 |
| %x,%X | 读入十六进制整数 |
| %s | 读入一个字符串，遇空格、制表符或换行符结束。 |
| %f,%F,%e,%E,%g,%G | 用来输入实数，可以用小数形式或指数形式输入。 |
| %p | 读入一个指针 |
| %u | 读入一个无符号十进制整数 |
| %n | 至此已读入值的等价字符数  scanf("%d%n",&a,&n); |
| %[] | 扫描字符集合，中括号中需要填写一个正则表达式，中括号内的内容不是以**^**开头的时候，表示只读取在中括号中出现的内容，当遇到第一个没有出现的字符时，就停止读取，并把目前已经读取的内容保存到对应的字符数组中；如果扫描集的第一个字符是**^**，那么读取规则就变成了只读取没有出现在扫描集中的字符，遇到第一个出现在扫描集中的字符时，读取即告停止，例如：  char a[80];scanf("%[0-9]",a); //只读取数字  scanf("%[^0-9]",a);// 只读取非数字字符 |
| %% | 读%符号 |

附加格式说明字符表修饰符说明 ：  
　　L/l 长度修饰符 输入"长"数据  
　　h 长度修饰符 输入"短"数据  
　　W 整型常数 指定输入数据所占宽度  
　　\* 表示本输入项在读入后不赋值给相应的变量

## int fflush( FILE \*stream );

链接： [fflush](http://msdn.microsoft.com/zh-cn/library/9yky46tz.aspx)

头文件：<stdio.h>

**功能：**fflush 函数刷新流。 如果关联 stream 的文件输出流，fflush 将与流关联的缓冲区内容写到文件中。 如果是输入流，fflush 清除缓冲区内容。此外，fflush(NULL) 刷新所有输出流。 流在调用之后保持打开。 fflush 不会影响未缓冲的流的效果。

**返回：**如果缓冲区刷新成功，则 fflush 返回 0。指定流没有缓冲区或只读的打开状态的情况下返回 值 0。 返回值 EOF 指示一个错误。

fflush(stdin);//这个方法在GCC下不可用。考虑移植则用下面的代码较好

void flush() {　　char c;　　while ((c=getchar()) != '/n'&&c!=EOF) ;　　}

## int feof( FILE \*stream );

链接： [feof](http://msdn.microsoft.com/zh-cn/library/xssktc6e.aspx)

头文件：<stdio.h>

**功能：**检测流上的文件结束符

**返回：**如果文件结束，则返回非0值，否则返回0，文件结束符只能被clearerr()清除。

FILE\*stream;

stream=fopen("DUMMY.FIL","r");

fgetc(stream);

if(feof(stream))

printf("Wehavereachedendoffile\n");

fclose(stream);

## int ferror( FILE \*stream );

链接： [ferror](http://msdn.microsoft.com/zh-cn/library/y2wc3w90.aspx)

头文件：<stdio.h>

**功能：**在流中测试错误

**返回：**如果stream没有发生错误，ferror 返回 0。 否则，它返回一个非零值。

#include <stdio.h>

#include <stdlib.h>

int main( void )

{

int count, total = 0;

char buffer[100];

FILE \*stream;

fopen\_s( &stream, "crt\_feof.txt", "r" );

if( stream == NULL )

exit( 1 );

// Cycle until end of file reached:

while( !feof( stream ) )

{

// Attempt to read in 100 bytes:

count = fread( buffer, sizeof( char ), 100, stream );

if( ferror( stream ) ) {

perror( "Read error" );

break;

}

// Total up actual bytes read

total += count;

}

printf( "Number of bytes read = %d\n", total );

fclose( stream );

}

## char \*gets( char \*buffer );

链接： [gets](http://msdn.microsoft.com/query/dev11.query?appId=Dev11IDEF1&l=ZH-CN&k=k(stdio%2Fgets);k(gets);k(DevLang-C%2B%2B);k(TargetOS-Windows)&rd=true)

头文件：<stdio.h>

**功能：** 从标准输入流 stdin 中读取一行并保存在 buffer中，读取’\n’，但是在返回行之前使用 null 字符 （’\0’ ）替换换行符

**返回：**读取成功的时候返回str，失败返回NULL 指针指向错误或文件结尾条件。使用 [ferror](http://msdn.microsoft.com/zh-cn/library/y2wc3w90.aspx) 或 [feof](http://msdn.microsoft.com/zh-cn/library/xssktc6e.aspx) 确定哪个已发生。

**注意：** 本函数可以无限读取，不会判断上限，所以程序员应该确保str的空间足够大，以便在执行读操作时不发生溢出。

## int puts(const char \*str );

链接： [puts](http://msdn.microsoft.com/zh-cn/library/tf52y4t1.aspx)

头文件：<stdio.h>

**功能：**puts 函数将str写入到标准输出流stdout，在输出流用一个换行符(‘\n’)替换字符串的终止 null 字符 (‘\0’)。

**返回：**如果成功返回非负值。 如果 puts 失败，则返回EOF

## char \*fgets(char \* str, int n, FILE \*stream );

链接： [fgets](http://msdn.microsoft.com/zh-cn/library/c37dh6kf.aspx)

头文件：<stdio.h>

**功能：**从文件结构体指针stream中读取数据，每次读取一行。读取的数据保存在str指向的字符数组中，每次最多读取n-1个字符（第n个字符赋'\0'），如果文件中的该行不足n个字符，则读完该行就结束。会读取’\n’

**返回：**读取成功的时候返回str，失败返回NULL 指针指向错误或文件结尾条件。使用 [ferror](http://msdn.microsoft.com/zh-cn/library/y2wc3w90.aspx) 或 [feof](http://msdn.microsoft.com/zh-cn/library/xssktc6e.aspx) 确定哪个已发生。

fgets( line, 100, stdin);

## int fputs( const char \*str, FILE \*stream );

链接：[fputs](http://msdn.microsoft.com/zh-cn/library/t33ya8ky.aspx)

头文件：<stdio.h>

**功能：** 不会添加换行符

**返回：**如果成功返回非负值。 如果 puts 失败，则返回EOF

fputs( "Hello world from fputs.\n", stdout );

## int fgetc( FILE \*stream );

链接： [fgetc](http://msdn.microsoft.com/zh-cn/library/c7sskzc1.aspx)

头文件：<stdio.h>

**功能：**读取文件的当前位置的单个字符， getc 与 fgetc等效，但是仅实现为函数，而不是函数和宏。

**返回：**fgetc 返回读取的字符或返回EOF 指示错误或文件结尾

**注意：**区别在于调用getc函数时所用的参数fp不能是有副作用的表达式，而fgetc函数则可以，也就是说，getc可以被当作宏来调用，而fgetc只能作为函数来调用。 因此，调用getc比调用fgetc耗费的时间少。

## int fputc( int c, FILE \*stream );

链接： [fputc](http://msdn.microsoft.com/zh-cn/library/yah67377.aspx)

头文件：<stdio.h>

**功能：**与 putc等效，但是仅实现为函数，而不是函数和宏。

**返回：**

## int getc( FILE \*stream );

链接：[getc](http://msdn.microsoft.com/zh-cn/library/5231d02a.aspx)

头文件：<stdio.h>

**功能：** 从流中读取一个字符

**返回：**返回读取的字符。 若要指示读取错误或文件结束的条件，getc 返回EOF

FILE\* fp; ch = getc(fp); ch = getc(stdin);

## int putc(int c,FILE \*stream );

链接：[putc](http://msdn.microsoft.com/zh-cn/library/0c8e91sd.aspx)

头文件：<stdio.h>

**功能：**

**返回：** 返回已写好的字符。 若要指示错误或文件结尾情况，putc 和 putchar 返回 EOF. 用 [ferror](http://msdn.microsoft.com/zh-cn/library/y2wc3w90.aspx) 或 [feof](http://msdn.microsoft.com/zh-cn/library/xssktc6e.aspx) 检测错误或文件结尾。

ch = putc( ‘a’, stdout );

## int getchar();

链接：[getchar](http://msdn.microsoft.com/query/dev11.query?appId=Dev11IDEF1&l=ZH-CN&k=k(stdio%2Fgetchar);k(getchar);k(DevLang-C%2B%2B);k(TargetOS-Windows)&rd=true)

头文件：<stdio.h>

**功能：** 输入一个字符

**返回：**返回读取的字符。请使用 **ferror** 或 **feof** 来检查是否有错误或文件结束。

ch = getchar();

## int putchar(int c );

链接：

头文件：<stdio.h>

**功能：** 输出一个字符

**返回：**返回已写好的字符。 若要指示错误或文件结尾条件，putc 和 putchar 返回 EOF;

## char \*strcpy(char \*strDestination, const char \*strSource );

链接：[strcpy](http://msdn.microsoft.com/zh-cn/library/kk6xf663.aspx)

头文件：<string.h>

**功能：**字符串复制

**返回：**返回一个目标字符串 (*strDestination*)。 没有保留任何返回值以指示错误

## char \*strncpy(char \*strDest, const char \*strSource, size\_t count );

链接： [strncpy](http://msdn.microsoft.com/zh-cn/library/xdsywd25.aspx)

头文件：<string.h>

**功能：**字符串复制，果 count 小于或等于 strSource的长度， null 字符不会自动附加于复制的字符串。

**返回：**返回一个目标字符串 (*strDestination*)。 没有保留任何返回值以指示错误

## char \*strcat(char \*strDestination, const char \*strSource);

链接：[strcat](http://msdn.microsoft.com/zh-cn/library/h1x0y282.aspx)

头文件：<string.h>

**功能：**将第二个字符串添加到第一个字符串后面

**返回：**返回一个目标字符串 (*strDestination*)。 没有保留任何返回值以指示错误

## char \*strncat(char \*strDest, const char \*strSource, size\_t count );

链接：[strncat](http://msdn.microsoft.com/zh-cn/library/tbyd7s1y.aspx)

头文件：<string.h>

**功能：**将第二个字符串添加到第一个字符串后面，count 限制追加的字符数；它不是限制 strDest的范围

**返回：**返回一个目标字符串 (*strDestination*)。 没有保留任何返回值以指示错误

## size\_t strlen( const char \*str);

链接： [strlen](http://msdn.microsoft.com/query/dev11.query?appId=Dev11IDEF1&l=ZH-CN&k=k(string%2Fstrlen);k(strlen);k(DevLang-C%2B%2B);k(TargetOS-Windows)&rd=true)

头文件：<string.h>

**功能：** strlen 将字符串解释为单字节字符字符串，因此其返回值与字节数总是相同的，即使该字符串包含一个多字节字符。

**返回：**返回str中的字符数目，不包括末端的 NULL。

## char \*strchr( const char \*str,int c ); // C only

链接： [strchr](http://msdn.microsoft.com/zh-cn/library/b34ccac3.aspx)

头文件：<string.h>

**功能：**在str中查找 c第一次出现的地方并返回位置，或者返回 NULL

**返回：**返回一个指针指向str中的 c第一次出现的位置或者，如果c没有被发现，会指向NULL

int ch = 'r';

char string[] = "The quick brown dog jumps over the lazy fox";

char \*pdest;

pdest = strchr( string, ch );

## char \*strrchr( const char \*str, int c ); // C only

链接： [strrchr](http://msdn.microsoft.com/zh-cn/library/ftw0heb9.aspx)

头文件：<string.h>

**功能：**在str中查找 c最后出现的地方并返回位置，或者返回 NULL

**返回：**Returns a pointer to the last occurrence of c in str, or NULL if c is not found.

## int strcmp( const char \*string1,const char \*string2);

链接： [strcmp](http://msdn.microsoft.com/zh-cn/library/e0z9k731.aspx)

头文件：<string.h>

**功能：**

**返回：**

## int strncmp( const char \*string1, const char \*string2, size\_t count );

链接： [strncmp](http://msdn.microsoft.com/zh-cn/library/eywx8zcx.aspx)

头文件：<string.h>

**功能：**比较前count个字符

**返回：**

## char \*strpbrk( const char \*str,const char \*strCharSet ); // C only

链接： [strpbrk](http://msdn.microsoft.com/query/dev11.query?appId=Dev11IDEF1&l=ZH-CN&k=k(strpbrk);k(DevLang-C%2B%2B);k(TargetOS-Windows)&rd=true) string pointer break

头文件：<string.h>

**功能：**依次检验字符串str中的字符，当被检验字符在字符串strCharSet中也包含时，则停止检验，并返回该字符位置，空字符NULL不包括在内。

**返回：**返回str中第一个满足条件的字符的指针，如果没有匹配字符则返回空指针NULL

char string[100] = "The 3 men and 2 boys ate 5 pigs\n";

char \*result;

调用：

result = strpbrk( string, "0123456789" );//在string中找出最先出现0~9中某一个字符的位置

printf( "1: %s\n", result++ );

result = strpbrk( result, "0123456789" );printf( "2: %s\n", result++ );

result = strpbrk( result, "0123456789" );printf( "3: %s\n", result );

## char \*strstr( const char \*str,const char \*strSearch ); // C only

链接： [strstr](http://msdn.microsoft.com/query/dev11.query?appId=Dev11IDEF1&l=ZH-CN&k=k(strstr);k(DevLang-C%2B%2B);k(TargetOS-Windows)&rd=true)

头文件：<string.h>

**功能：**搜索strSearch在str中的第一次出现

**返回：**该函数返回strSearch第一次在str中的位置，如果没有找到，返回NULL

char str[]="1234xyz";

char \*str1=strstr(str,"34");

## int memcmp( const void \*buf1,const void \*buf2,size\_t count);

链接： [memcmp](http://msdn.microsoft.com/zh-cn/library/zyaebf12.aspx)

头文件：<string.h>或者<memory.h>

**功能：**在两种缓冲区中比较字符。

**返回：**

## int sprintf(char \*buffer,const char \*format [,argument] ...);

链接： [sprintf](http://msdn.microsoft.com/zh-cn/library/ybk95axf.aspx)

头文件：<stdio.h>

**功能：**将格式数据写入字符串

**返回：**sprintf 返回存储在 buffer 里的字节数，不包括终止 null 字符。buffer 或 format 不能是null指针，

## int sscanf(const char \*buffer,const char \*format [,argument ] ... );

链接： [sscanf](http://msdn.microsoft.com/zh-cn/library/zkx076cy.aspx)

头文件：<stdio.h>

**功能：**从字符串中读取格式化数据

**返回：**每个函数返回成功转换并分配的字段数量；返回值不包括已读取但未分配的字段。 返回值为 0 表示未分配字段。 如果出现错误，或者，如果在第一个转换之前到达字符串的末尾，则返回值是 EOF。

## int fprintf( FILE \*stream,const char \*format [,argument ]...);

链接： [fprintf](http://msdn.microsoft.com/zh-cn/library/xkh07fe2.aspx)

头文件：<stdio.h>

**功能：**fprintf 格式化并输出一系列字符和值到输出stream.

**返回：**fprintf 返回写入的字节数

## int fscanf( FILE \*stream,const char \*format [,argument ]... );

链接： [fscanf](http://msdn.microsoft.com/zh-cn/library/cb5kac8b.aspx)

头文件：<stdio.h>

**功能：**从 stream 当前位置中读取数据到由 argument 参数列表（如果有）给定的位置中

**返回：**每个函数返回成功转换并分配的字段数量；返回值不包括已读取但未分配的字段。 返回值为 0 表示未分配字段。 如果出现错误，或者，如果在第一个转换之前到达文件流的末尾，则返回值是EOF。