**C 标准库 - <limits.h>**

**简介**

**limits.h** 头文件决定了各种变量类型的各种属性。定义在该头文件中的宏限制了各种变量类型（比如 char、int 和 long）的值。

这些限制指定了变量不能存储任何超出这些限制的值，例如一个无符号可以存储的最大值是 255。

**库宏**

下面的值是特定实现的，且是通过 #define 指令来定义的，这些值都不得低于下边所给出的值。

|  |  |  |
| --- | --- | --- |
| **宏** | **值** | **描述** |
| CHAR\_BIT | 8 | 定义一个字节的比特数。 |
| SCHAR\_MIN | -128 | 定义一个有符号字符的最小值。 |
| SCHAR\_MAX | 127 | 定义一个有符号字符的最大值。 |
| UCHAR\_MAX | 255 | 定义一个无符号字符的最大值。 |
| CHAR\_MIN | 0 | 定义类型 char 的最小值，如果 char 表示负值，则它的值等于 SCHAR\_MIN，否则等于 0。 |
| CHAR\_MAX | 127 | 定义类型 char 的最大值，如果 char 表示负值，则它的值等于 SCHAR\_MAX，否则等于 UCHAR\_MAX。 |
| MB\_LEN\_MAX | 1 | 定义多字节字符中的最大字节数。 |
| SHRT\_MIN | -32768 | 定义一个短整型的最小值。 |
| SHRT\_MAX | +32767 | 定义一个短整型的最大值。 |
| USHRT\_MAX | 65535 | 定义一个无符号短整型的最大值。 |
| INT\_MIN | -32768 | 定义一个整型的最小值。 |
| INT\_MAX | +32767 | 定义一个整型的最大值。 |
| UINT\_MAX | 65535 | 定义一个无符号整型的最大值。 |
| LONG\_MIN | -2147483648 | 定义一个长整型的最小值。 |
| LONG\_MAX | +2147483647 | 定义一个长整型的最大值。 |
| ULONG\_MAX | 4294967295 | 定义一个无符号长整型的最大值。 |

**实例**

下面的实例演示了 limit.h 文件中定义的一些常量的使用。

#include <stdio.h>

#include <limits.h>

int main()

{

printf("The number of bits in a byte %d\n", CHAR\_BIT);

printf("The minimum value of SIGNED CHAR = %d\n", SCHAR\_MIN);

printf("The maximum value of SIGNED CHAR = %d\n", SCHAR\_MAX);

printf("The maximum value of UNSIGNED CHAR = %d\n", UCHAR\_MAX);

printf("The minimum value of SHORT INT = %d\n", SHRT\_MIN);

printf("The maximum value of SHORT INT = %d\n", SHRT\_MAX);

printf("The minimum value of INT = %d\n", INT\_MIN);

printf("The maximum value of INT = %d\n", INT\_MAX);

printf("The minimum value of CHAR = %d\n", CHAR\_MIN);

printf("The maximum value of CHAR = %d\n", CHAR\_MAX);

printf("The minimum value of LONG = %ld\n", LONG\_MIN);

printf("The maximum value of LONG = %ld\n", LONG\_MAX);

return(0);

}

让我们编译和运行上面的程序，这将产生下列结果：

The number of bits in a byte 8

The minimum value of SIGNED CHAR = -128

The maximum value of SIGNED CHAR = 127

The maximum value of UNSIGNED CHAR = 255

The minimum value of SHORT INT = -32768

The maximum value of SHORT INT = 32767

The minimum value of INT = -32768

The maximum value of INT = 32767

The minimum value of CHAR = -128

The maximum value of CHAR = 127

The minimum value of LONG = -2147483648

The maximum value of LONG = 2147483647

**C 标准库 - <string.h>**

**简介**

**string .h** 头文件定义了一个变量类型、一个宏和各种操作字符数组的函数。

**库变量**

下面是头文件 string.h 中定义的变量类型：

|  |  |
| --- | --- |
| **序号** | **变量 & 描述** |
| 1 | **size\_t** 这是无符号整数类型，它是 **sizeof** 关键字的结果。 |

**库宏**

下面是头文件 string.h 中定义的宏：

|  |  |
| --- | --- |
| **序号** | **宏 & 描述** |
| 1 | **NULL** 这个宏是一个空指针常量的值。 |

**库函数**

下面是头文件 string.h 中定义的函数：

|  |  |
| --- | --- |
| **序号** | **函数 & 描述** |
| 1 | [void \*memchr(const void \*str, int c, size\_t n)](http://www.runoob.com/cprogramming/c-function-memchr.html) 在参数 *str* 所指向的字符串的前 n 个字节中搜索第一次出现字符 c（一个无符号字符）的位置。 |
| 2 | [int memcmp(const void \*str1, const void \*str2, size\_t n)](http://www.runoob.com/cprogramming/c-function-memcmp.html) 把 *str1* 和 *str2* 的前 n 个字节进行比较。 |
| 3 | [void \*memcpy(void \*dest, const void \*src, size\_t n)](http://www.runoob.com/cprogramming/c-function-memcpy.html) 从 src 复制 n 个字符到 *dest*。 |
| 4 | [void \*memmove(void \*dest, const void \*src, size\_t n)](http://www.runoob.com/cprogramming/c-function-memmove.html) 另一个用于从 *str2* 复制 n 个字符到 *str1* 的函数。 |
| 5 | [void \*memset(void \*str, int c, size\_t n)](http://www.runoob.com/cprogramming/c-function-memset.html) 复制字符 c（一个无符号字符）到参数 *str* 所指向的字符串的前 n 个字符。 |
| 6 | [char \*strcat(char \*dest, const char \*src)](http://www.runoob.com/cprogramming/c-function-strcat.html) 把 *src* 所指向的字符串追加到 *dest* 所指向的字符串的结尾。 |
| 7 | [char \*strncat(char \*dest, const char \*src, size\_t n)](http://www.runoob.com/cprogramming/c-function-strncat.html) 把 *src* 所指向的字符串追加到 *dest* 所指向的字符串的结尾，直到 n 字符长度为止。 |
| 8 | [char \*strchr(const char \*str, int c)](http://www.runoob.com/cprogramming/c-function-strchr.html) 在参数 *str* 所指向的字符串中搜索第一次出现字符 c（一个无符号字符）的位置。 |
| 9 | [int strcmp(const char \*str1, const char \*str2)](http://www.runoob.com/cprogramming/c-function-strcmp.html) 把 *str1* 所指向的字符串和 *str2* 所指向的字符串进行比较。 |
| 10 | [int strncmp(const char \*str1, const char \*str2, size\_t n)](http://www.runoob.com/cprogramming/c-function-strncmp.html) 把 *str1* 和 *str2* 进行比较，最多比较前 n 个字节。 |
| 11 | [int strcoll(const char \*str1, const char \*str2)](http://www.runoob.com/cprogramming/c-function-strcoll.html) 把 *str1* 和 *str2* 进行比较，结果取决于 LC\_COLLATE 的位置设置。 |
| 12 | [char \*strcpy(char \*dest, const char \*src)](http://www.runoob.com/cprogramming/c-function-strcpy.html) 把 *src* 所指向的字符串复制到 *dest*。 |
| 13 | [char \*strncpy(char \*dest, const char \*src, size\_t n)](http://www.runoob.com/cprogramming/c-function-strncpy.html) 把 *src* 所指向的字符串复制到 *dest*，最多复制 n 个字符。 |
| 14 | [size\_t strcspn(const char \*str1, const char \*str2)](http://www.runoob.com/cprogramming/c-function-strcspn.html) 检索字符串 str1 开头连续有几个字符都不含字符串 str2 中的字符。 |
| 15 | [char \*strerror(int errnum)](http://www.runoob.com/cprogramming/c-function-strerror.html) 从内部数组中搜索错误号 errnum，并返回一个指向错误消息字符串的指针。 |
| 16 | [size\_t strlen(const char \*str)](http://www.runoob.com/cprogramming/c-function-strlen.html) 计算字符串 str 的长度，直到空结束字符，但不包括空结束字符。 |
| 17 | [char \*strpbrk(const char \*str1, const char \*str2)](http://www.runoob.com/cprogramming/c-function-strpbrk.html) 检索字符串 *str1* 中第一个匹配字符串 *str2* 中字符的字符，不包含空结束字符。也就是说，依次检验字符串 str1 中的字符，当被检验字符在字符串 str2 中也包含时，则停止检验，并返回该字符位置。 |
| 18 | [char \*strrchr(const char \*str, int c)](http://www.runoob.com/cprogramming/c-function-strrchr.html) 在参数 *str* 所指向的字符串中搜索最后一次出现字符 c（一个无符号字符）的位置。 |
| 19 | [size\_t strspn(const char \*str1, const char \*str2)](http://www.runoob.com/cprogramming/c-function-strspn.html) 检索字符串 *str1* 中第一个不在字符串 *str2* 中出现的字符下标。 |
| 20 | [char \*strstr(const char \*haystack, const char \*needle)](http://www.runoob.com/cprogramming/c-function-strstr.html) 在字符串 *haystack* 中查找第一次出现字符串 *needle*（不包含空结束字符）的位置。 |
| 21 | [char \*strtok(char \*str, const char \*delim)](http://www.runoob.com/cprogramming/c-function-strtok.html) 分解字符串 *str* 为一组字符串，*delim* 为分隔符。 |
| 22 | [size\_t strxfrm(char \*dest, const char \*src, size\_t n)](http://www.runoob.com/cprogramming/c-function-strxfrm.html) 根据程序当前的区域选项中的 LC\_COLLATE 来转换字符串 **src** 的前 **n** 个字符，并把它们放置在字符串 **dest** 中。 |

**C 标准库 - <stdlib.h>**

**简介**

**stdlib .h** 头文件定义了四个变量类型、一些宏和各种通用工具函数。

**库变量**

下面是头文件 stdlib.h 中定义的变量类型：

|  |  |
| --- | --- |
| **序号** | **变量 & 描述** |
| 1 | **size\_t** 这是无符号整数类型，它是 **sizeof** 关键字的结果。 |
| 2 | **wchar\_t** 这是一个宽字符常量大小的整数类型。 |
| 3 | **div\_t** 这是 **div** 函数返回的结构。 |
| 4 | **ldiv\_t** 这是 **ldiv** 函数返回的结构。 |

**库宏**

下面是头文件 stdlib.h 中定义的宏：

|  |  |
| --- | --- |
| **序号** | **宏 & 描述** |
| 1 | **NULL** 这个宏是一个空指针常量的值。 |
| 2 | **EXIT\_FAILURE** 这是 exit 函数失败时要返回的值。 |
| 3 | **EXIT\_SUCCESS** 这是 exit 函数成功时要返回的值。 |
| 4 | **RAND\_MAX** 这个宏是 rand 函数返回的最大值。 |
| 5 | **MB\_CUR\_MAX** 这个宏表示在多字节字符集中的最大字符数，不能大于 MB\_LEN\_MAX。 |

**库函数**

下面是头文件 stdlib.h 中定义的函数：

|  |  |
| --- | --- |
| **序号** | **函数 & 描述** |
| 1 | [double atof(const char \*str)](http://www.runoob.com/cprogramming/c-function-atof.html) 把参数 *str* 所指向的字符串转换为一个浮点数（类型为 double 型）。 |
| 2 | [int atoi(const char \*str)](http://www.runoob.com/cprogramming/c-function-atoi.html) 把参数 *str* 所指向的字符串转换为一个整数（类型为 int 型）。 |
| 3 | [long int atol(const char \*str)](http://www.runoob.com/cprogramming/c-function-atol.html) 把参数 *str* 所指向的字符串转换为一个长整数（类型为 long int 型）。 |
| 4 | [double strtod(const char \*str, char \*\*endptr)](http://www.runoob.com/cprogramming/c-function-strtod.html) 把参数 *str* 所指向的字符串转换为一个浮点数（类型为 double 型）。 |
| 5 | [long int strtol(const char \*str, char \*\*endptr, int base)](http://www.runoob.com/cprogramming/c-function-strtol.html) 把参数 *str* 所指向的字符串转换为一个长整数（类型为 long int 型）。 |
| 6 | [unsigned long int strtoul(const char \*str, char \*\*endptr, int base)](http://www.runoob.com/cprogramming/c-function-strtoul.html) 把参数 *str* 所指向的字符串转换为一个无符号长整数（类型为 unsigned long int 型）。 |
| 7 | [void \*calloc(size\_t nitems, size\_t size)](http://www.runoob.com/cprogramming/c-function-calloc.html) 分配所需的内存空间，并返回一个指向它的指针。 |
| 8 | [void free(void \*ptr)](http://www.runoob.com/cprogramming/c-function-free.html) 释放之前调用 *calloc、malloc* 或 *realloc* 所分配的内存空间。 |
| 9 | [void \*malloc(size\_t size)](http://www.runoob.com/cprogramming/c-function-malloc.html) 分配所需的内存空间，并返回一个指向它的指针。 |
| 10 | [void \*realloc(void \*ptr, size\_t size)](http://www.runoob.com/cprogramming/c-function-realloc.html) 尝试重新调整之前调用 *malloc* 或 *calloc* 所分配的 ptr 所指向的内存块的大小。 |
| 11 | [void abort(void)](http://www.runoob.com/cprogramming/c-function-abort.html) 使一个异常程序终止。 |
| 12 | [int atexit(void (\*func)(void))](http://www.runoob.com/cprogramming/c-function-atexit.html) 当程序正常终止时，调用指定的函数 **func**。 |
| 13 | [void exit(int status)](http://www.runoob.com/cprogramming/c-function-exit.html) 使程序正常终止。 |
| 14 | [char \*getenv(const char \*name)](http://www.runoob.com/cprogramming/c-function-getenv.html) 搜索 name 所指向的环境字符串，并返回相关的值给字符串。 |
| 15 | [int system(const char \*string)](http://www.runoob.com/cprogramming/c-function-system.html) 由 string 指定的命令传给要被命令处理器执行的主机环境。 |
| 16 | [void \*bsearch(const void \*key, const void \*base, size\_t nitems, size\_t size, int (\*compar)(const void \*, const void \*))](http://www.runoob.com/cprogramming/c-function-bsearch.html) 执行二分查找。 |
| 17 | [void qsort(void \*base, size\_t nitems, size\_t size, int (\*compar)(const void \*, const void\*))](http://www.runoob.com/cprogramming/c-function-qsort.html) 数组排序。 |
| 18 | [int abs(int x)](http://www.runoob.com/cprogramming/c-function-abs.html) 返回 x 的绝对值。 |
| 19 | [div\_t div(int numer, int denom)](http://www.runoob.com/cprogramming/c-function-div.html) 分子除以分母。 |
| 20 | [long int labs(long int x)](http://www.runoob.com/cprogramming/c-function-labs.html) 返回 x 的绝对值。 |
| 21 | [ldiv\_t ldiv(long int numer, long int denom)](http://www.runoob.com/cprogramming/c-function-ldiv.html) 分子除以分母。 |
| 22 | [int rand(void)](http://www.runoob.com/cprogramming/c-function-rand.html) 返回一个范围在 0 到 *RAND\_MAX* 之间的伪随机数。 |
| 23 | [void srand(unsigned int seed)](http://www.runoob.com/cprogramming/c-function-srand.html) 该函数播种由函数 **rand** 使用的随机数发生器。 |
| 24 | [int mblen(const char \*str, size\_t n)](http://www.runoob.com/cprogramming/c-function-mblen.html) 返回参数 *str* 所指向的多字节字符的长度。 |
| 25 | [size\_t mbstowcs(schar\_t \*pwcs, const char \*str, size\_t n)](http://www.runoob.com/cprogramming/c-function-mbstowcs.html) 把参数 *str* 所指向的多字节字符的字符串转换为参数 *pwcs* 所指向的数组。 |
| 26 | [int mbtowc(whcar\_t \*pwc, const char \*str, size\_t n)](http://www.runoob.com/cprogramming/c-function-mbtowc.html) 检查参数 *str* 所指向的多字节字符。 |
| 27 | [size\_t wcstombs(char \*str, const wchar\_t \*pwcs, size\_t n)](http://www.runoob.com/cprogramming/c-function-wcstombs.html) 把数组 *pwcs* 中存储的编码转换为多字节字符，并把它们存储在字符串 *str* 中。 |
| 28 | [int wctomb(char \*str, wchar\_t wchar)](http://www.runoob.com/cprogramming/c-function-wctomb.html) 检查对应于参数 *wchar* 所给出的多字节字符的编码。 |

**C 标准库 - <ctype.h>**

**简介**

C 标准库的 **ctype.h** 头文件提供了一些函数，可用于测试和映射字符。

这些函数接受 **int** 作为参数，它的值必须是 EOF 或表示为一个无符号字符。

如果参数 c 满足描述的条件，则这些函数返回非零（true）。如果参数 c 不满足描述的条件，则这些函数返回零。

**库函数**

下面列出了头文件 ctype.h 中定义的函数：

|  |  |
| --- | --- |
| **序号** | **函数 & 描述** |
| 1 | [int isalnum(int c)](http://www.runoob.com/cprogramming/c-function-isalnum.html) 该函数检查所传的字符是否是字母和数字。 |
| 2 | [int isalpha(int c)](http://www.runoob.com/cprogramming/c-function-isalpha.html) 该函数检查所传的字符是否是字母。 |
| 3 | [int iscntrl(int c)](http://www.runoob.com/cprogramming/c-function-iscntrl.html) 该函数检查所传的字符是否是控制字符。 |
| 4 | [int isdigit(int c)](http://www.runoob.com/cprogramming/c-function-isdigit.html) 该函数检查所传的字符是否是十进制数字。 |
| 5 | [int isgraph(int c)](http://www.runoob.com/cprogramming/c-function-isgraph.html) 该函数检查所传的字符是否有图形表示法。 |
| 6 | [int islower(int c)](http://www.runoob.com/cprogramming/c-function-islower.html) 该函数检查所传的字符是否是小写字母。 |
| 7 | [int isprint(int c)](http://www.runoob.com/cprogramming/c-function-isprint.html) 该函数检查所传的字符是否是可打印的。 |
| 8 | [int ispunct(int c)](http://www.runoob.com/cprogramming/c-function-ispunct.html) 该函数检查所传的字符是否是标点符号字符。 |
| 9 | [int isspace(int c)](http://www.runoob.com/cprogramming/c-function-isspace.html) 该函数检查所传的字符是否是空白字符。 |
| 10 | [int isupper(int c)](http://www.runoob.com/cprogramming/c-function-isupper.html) 该函数检查所传的字符是否是大写字母。 |
| 11 | [int isxdigit(int c)](http://www.runoob.com/cprogramming/c-function-isxdigit.html) 该函数检查所传的字符是否是十六进制数字。 |

标准库还包含了两个转换函数，它们接受并返回一个 "int"

|  |  |
| --- | --- |
| **序号** | **函数 & 描述** |
| 1 | [int tolower(int c)](http://www.runoob.com/cprogramming/c-function-tolower.html) 该函数把大写字母转换为小写字母。 |
| 2 | [int toupper(int c)](http://www.runoob.com/cprogramming/c-function-toupper.html) 该函数把小写字母转换为大写字母。 |

**字符类**

|  |  |
| --- | --- |
| **序号** | **字符类 & 描述** |
| 1 | **数字** 完整的数字集合 { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 } |
| 2 | **十六进制数字** 集合 { 0 1 2 3 4 5 6 7 8 9 A B C D E F a b c d e f } |
| 3 | **小写字母** 集合 { a b c d e f g h i j k l m n o p q r s t u v w x y z } |
| 4 | **大写字母** 集合 {A B C D E F G H I J K L M N O P Q R S T U V W X Y Z } |
| 5 | **字母** 小写字母和大写字母的集合 |
| 6 | **字母数字字符** 数字、小写字母和大写字母的集合 |
| 7 | **标点符号字符** 集合 ! " # $ % & ' ( ) \* + , - . / : ; < = > ? @ [ \ ] ^ \_ ` { | } ~ |
| 8 | **图形字符** 字母数字字符和标点符号字符的集合 |
| 9 | **空格字符** 制表符、换行符、垂直制表符、换页符、回车符、空格符的集合。 |
| 10 | **可打印字符** 字母数字字符、标点符号字符和空格字符的集合。 |
| 11 | **控制字符** 在 ASCII 编码中，这些字符的八进制代码是从 000 到 037，以及 177（DEL）。 |
| 12 | **空白字符** 包括空格符和制表符。 |
| 13 | **字母字符** 小写字母和大写字母的集合。 |

**C 标准库 - <math.h>**

**简介**

**math.h** 头文件定义了各种数学函数和一个宏。在这个库中所有可用的功能都带有一个 **double** 类型的参数，且都返回 **double**类型的结果。

**库宏**

下面是这个库中定义的唯一的一个宏：

|  |  |
| --- | --- |
| **序号** | **宏 & 描述** |
| 1 | **HUGE\_VAL**  当函数的结果不可以表示为浮点数时。如果是因为结果的幅度太大以致于无法表示，则函数会设置 errno 为 ERANGE 来表示范围错误，并返回一个由宏 HUGE\_VAL 或者它的否定（- HUGE\_VAL）命名的一个特定的很大的值。  如果结果的幅度太小，则会返回零值。在这种情况下，error 可能会被设置为 ERANGE，也有可能不会被设置为 ERANGE。 |

**库函数**

下面列出了头文件 math.h 中定义的函数：

|  |  |
| --- | --- |
| **序号** | **函数 & 描述** |
| 1 | [double acos(double x)](http://www.runoob.com/cprogramming/c-function-acos.html) 返回以弧度表示的 x 的反余弦。 |
| 2 | [double asin(double x)](http://www.runoob.com/cprogramming/c-function-asin.html) 返回以弧度表示的 x 的反正弦。 |
| 3 | [double atan(double x)](http://www.runoob.com/cprogramming/c-function-atan.html) 返回以弧度表示的 x 的反正切。 |
| 4 | [double atan2(double y, double x)](http://www.runoob.com/cprogramming/c-function-atan2.html) 返回以弧度表示的 y/x 的反正切。y 和 x 的值的符号决定了正确的象限。 |
| 5 | [double cos(double x)](http://www.runoob.com/cprogramming/c-function-cos.html) 返回弧度角 x 的余弦。 |
| 6 | [double cosh(double x)](http://www.runoob.com/cprogramming/c-function-cosh.html) 返回 x 的双曲余弦。 |
| 7 | [double sin(double x)](http://www.runoob.com/cprogramming/c-function-sin.html) 返回弧度角 x 的正弦。 |
| 8 | [double sinh(double x)](http://www.runoob.com/cprogramming/c-function-sinh.html) 返回 x 的双曲正弦。 |
| 9 | [double tanh(double x)](http://www.runoob.com/cprogramming/c-function-tanh.html) 返回 x 的双曲正切。 |
| 10 | [double exp(double x)](http://www.runoob.com/cprogramming/c-function-exp.html) 返回 e 的 x 次幂的值。 |
| 11 | [double frexp(double x, int \*exponent)](http://www.runoob.com/cprogramming/c-function-frexp.html) 把浮点数 x 分解成尾数和指数。返回值是尾数，并将指数存入 exponent 中。所得的值是 x = mantissa \* 2 ^ exponent。 |
| 12 | [double ldexp(double x, int exponent)](http://www.runoob.com/cprogramming/c-function-ldexp.html) 返回 x 乘以 2 的 exponent 次幂。 |
| 13 | [double log(double x)](http://www.runoob.com/cprogramming/c-function-log.html) 返回 x 的自然对数（基数为 e 的对数）。 |
| 14 | [double log10(double x)](http://www.runoob.com/cprogramming/c-function-log10.html) 返回 x 的常用对数（基数为 10 的对数）。 |
| 15 | [double modf(double x, double \*integer)](http://www.runoob.com/cprogramming/c-function-modf.html) 返回值为小数部分（小数点后的部分），并设置 integer 为整数部分。 |
| 16 | [double pow(double x, double y)](http://www.runoob.com/cprogramming/c-function-pow.html) 返回 x 的 y 次幂。 |
| 17 | [double sqrt(double x)](http://www.runoob.com/cprogramming/c-function-sqrt.html) 返回 x 的平方根。 |
| 18 | [double ceil(double x)](http://www.runoob.com/cprogramming/c-function-ceil.html) 返回大于或等于 x 的最小的整数值。 |
| 19 | [double fabs(double x)](http://www.runoob.com/cprogramming/c-function-fabs.html) 返回 x 的绝对值。 |
| 20 | [double floor(double x)](http://www.runoob.com/cprogramming/c-function-floor.html) 返回小于或等于 x 的最大的整数值。 |
| 21 | [double fmod(double x, double y)](http://www.runoob.com/cprogramming/c-function-fmod.html) 返回 x 除以 y 的余数。 |