

附录 常用库函数

常用函数库列表

序号	函数库	功 能
1	stdio.h	输入输出函数库
2	math.h	数学函数库
3	string.h	字符串处理函数库
4	stdlib.h	辅助处理函数库
5	time.h	时间函数库
6	conio.h	控制台输入输出函数库
7	ctype.h	字符类型函数库
8	winbase.h	Windows 基础函数库

1. 输入输出函数库，#include <stdio.h>

(1) 标准输入输出类

类别	函 数 声 明	功 能
常 规	int printf (const char *format, ...);	按格式化串 format 要求，将格式化后的数据流输出到标准输出设备 stdout 上，成功则返回输出的字符数，失败则返回 EOF (-1)。
	int scanf (const char *format, ...);	从标准输入设备 stdin 上格式化输入数据，返回输入的数据个数。从键盘上输入时，输入内容同时回显在屏幕上，并在按下回车后才开始处理。
	int putchar (int c);	输出字符 c 到标准输出设备 stdout 上，返回字符 c。
	int getchar (void);	从标准输入设备 stdin 上输入一个字符并返回。
	int puts (const char *s);	输出字符串 s 到标准输出设备 stdout 上，并换行。
	char * gets (char *s);	从标准输入设备 stdin 上输入一行字符串，返回 s。
字 符 串 流	int sprintf (char *s, const char *format, ...);	按格式化串 format 要求，将格式化后的数据流输出到字符串缓冲区 s 中，成功则返回输出的字符数，失败则返回 EOF (-1)。
	int sscanf (const char *s, const char *format, ...);	从字符串缓冲区 s 上格式化输入数据，返回输入的数据个数。

(2) 文件流读写类

类别	函 数 声 明	功 能
文 件 常 规	FILE * fopen (const char *filename, const char *type);	以模式 type 打开文件 filename，成功则返回文件流指针，失败则返回 NULL。
	int fclose (FILE *fp);	关闭由 fopen 打开的文件，成功返回 0，失败返回 EOF (-1)。
	int fprintf (FILE *fp, const char *format, ...);	按格式化串 format 要求，将格式化后的数据流输出到文件流 fp 中，成功则返回输出的字符数，失败则返回 EOF (-1)。
	int fscanf (FILE *fp, const char *format, ...);	从文件流 fp 中格式化输入数据，返回输入的数据个数。
	int fputc (int c, FILE *fp);	输出字符 c 到文件流 fp 上，返回字符 c，如果失败返回 EOF (-1)。

	<code>int fgetc(FILE *fp);</code>	从文件流 fp 上输入一个字符并返回，如果文件读取失败或到达文件末尾，返回 EOF。
	<code>int fputs(const char *s, FILE *fp);</code>	输出字符串 s 到文件流 fp 上，并换行，成功则返回非负值，失败则返回 EOF。
	<code>char * fgets(char *s, int n, FILE *fp);</code>	从文件流 fp 上输入一行字符串，成功则返回 s，出错或到达文件结束，返回 NULL。
二进制	<code>int fwrite(const void *buff, int size, int n, FILE *fp);</code>	向文件流 fp 中写入 n 个长度为 size 字节的二进制数据，数据起始地址在 buff 中。成功返回实际写入数据项数，失败返回。
	<code>int fread(void *buff, int size, int n, FILE *fp)</code>	从文件流 fp 中读入 n 个长度为 size 字节的二进制数据到 buff 中，返回读取的数据项数。
定位类	<code>int feof(FILE *fp);</code>	检查文件是否到达末尾，文件结束返回非 0 值，否则返回 0。
	<code>long ftell(FILE *fp);</code>	确定文件所在位置，返回文件当前位置（距离文件起点的字节数），如果失败，返回 EOF。
	<code>int fseek(FILE *fp, long offset, int from);</code>	移动文件流 fp 的当前位置到新位置。from 取 SEEK_SET 或 0 时，以文件开头为基准，取 SEEK_CUR 或 1 时，以当前位置为基准，取 SEEK_END 或 2 时，以文件结尾为基准，offset 为从选定基准开始的正负偏移量。成功返回 0，失败返回非 0 值。
	<code>void rewind(FILE *fp);</code>	重置文件流当前位置，回到文件开头。
其他	<code>int fflush(FILE *fp);</code>	清除文件流 fp 缓冲区，如果文件写，所有写操作立即执行。成功返回 0，失败返回 EOF。
	<code>int ferror(FILE *fp);</code>	文件操作失败时，返回文件错误号，返回 0 表示没有错误。
	<code>void clearerr(FILE *fp);</code>	清除文件流 fp 的错误信息。

2. 数学函数库，#include <math.h>

类别	函数声明	功能
指数类	<code>double sqrt(double x);</code>	计算平方根
	<code>double exp(double x);</code>	指数函数，即 e^x
	<code>double pow(double x, double y);</code>	幂函数，即 x^y
	<code>double log(double x);</code>	对数函数 $\ln(x)$ ，以 e 为底
	<code>double log10(double x);</code>	对数函数 $\log_{10}(x)$ ，以 10 为底
三角类	<code>double sin(double x);</code>	正弦函数
	<code>double cos(double x);</code>	余弦函数
	<code>double tan(double x);</code>	正切函数
	<code>double asin(double x);</code>	反正弦函数
	<code>double acos(double x);</code>	反余弦函数
	<code>double atan(double x);</code>	反正切函数
	<code>double sinh(double x);</code>	双曲正弦函数
	<code>double cosh(double x);</code>	双曲余弦函数
其他	<code>double tanh(double x);</code>	双曲正切函数
	<code>double fabs(double x);</code>	返回浮点数的绝对值
	<code>double floor(double x);</code>	向下取整，返回 $\leq x$ 的最大整数
	<code>double ceil(double x);</code>	向上取整，返回 $\geq x$ 的最小整数
	<code>double fmod(double x, double y);</code>	计算 x 对 y 的模，即 x 除 y 的余数。

3. 字符串处理函数库, #include <string.h>

类别	函数声明	功能
常用	int strlen (const char *s);	返回字符串 s 的长度
	char * strcpy (char *dest, const char *src);	将字符串 src 复制到 dest, 返回 dest。
	char * strcat (char *dest, const char *src);	将字符串 src 添加到 dest 末尾, 返回 dest。
	int strcmp (const char *s1, const char *s2);	比较字符串 s1 与 s2 在字典中的先后顺序, 如果 s1 在前返回一个负数, 如果 s1 在后返回一个正数, 如果 s1 与 s2 完全相同, 返回 0。
拓展	int stricmp (const char *s1, const char *s2);	功能类似 strcmp, 但不区分大小写字母。
	char * strlwr (char *s);	将字符串 s 中的大写字母全部转换成小写字母, 返回转换后的字符串 s。
	char * strupr (char *s);	将字符串 s 中的小写字母全部转换成大写字母, 返回转换后的字符串 s。
	char * strrev (char *s);	将字符串 s 中的字符全部颠倒顺序重新排列, 返回排列后的字符串 s。
	char * strset (char *s, int c);	把字符串 s 中的所有字符都设置成字符 c, 返回设置后的字符串 s。
	char * strchr (const char *s, int c);	在字符串 s 中查找字符 c, 如果找到, 返回 s 中首次出现 c 的指针, 如果没有找到, 返回 NULL。
	char * strstr (const char *s1, const char *s2);	在字符串 s1 中查找子串 s2, 如果找到, 返回 s1 中首次出现 s2 的指针, 如果没有找到, 返回 NULL。
	int strspn (const char *s, const char *a);	扫描字符串 s, 返回字符串中第一个不在字符串 a 中出现的字符下标。用于过滤特定的字符集, 如 strspn(s, " \t\n") 返回第一个非空字符的下标。
	char * strtok (char *s1, const char *s2);	用于将字符串 s1 拆分成多个子串, s2 为子串之间允许的分隔符集, 当 s1 不为 NULL 时启动一轮新的拆分过程, 并返回拆分出的第一个子串, 当 s1 为 NULL 时延续上一轮拆分, 返回下一个子串, 返回值为 NULL 时表示所有子串拆分完成。 如先调用 p=strtok(s, " \t\n"), 再多次调用 p=strtok(NULL, " \t\n"), 则 p 依次指向字符串 s 中以空格、TAB 或换行作为分隔符的各个“单词”。

4. 辅助函数库, #include <stdlib.h>

类别	函数声明	功能
分配类	void * malloc (int size);	分配连续 size 字节的内存空间, 并返回首地址, 如果系统没有足够的内存空间, 返回 NULL。
	void free (void *buff);	释放已分配的内存空间。
	void * realloc (void *buff, int size);	申请重新分配内存, 调整原分配的 buff 空间为 size 字节。如果成功返回新分配空间的指针, 否则返回 NULL。
	void * calloc (int n, int size);	类似 malloc, 分配 n 个 size 字节的内存空间, 成功后该空间二进制清零。

随机数	<code>void srand(unsigned seed);</code>	设置伪随机数生成种子，种子不同，随后调用 <code>rand</code> 产生的伪随机数序列也不同。为了更好地达到“随机”效果，可以使用程序运行时的当前时间作为种子，即调用 <code>srand(time(NULL))</code> ;
	<code>int rand(void);</code>	返回下一个伪随机数，伪随机数取值 0 至 <code>RAND_MAX</code> (32767) 之间（含）。
系统类	<code>void exit(int ret);</code>	结束当前程序，并返回码 <code>ret</code> 。 <code>exit</code> 函数可以在任何函数中调用，在 <code>main</code> 函数中的作用等同于 <code>return(ret)</code> 。
	<code>int system(const char *command);</code>	执行一条命令行命令，并等待该命令执行完成。在 Windows 系统下，调用 <code>system("cls")</code> ; 可以清除程序运行窗口的屏幕。
	<code>char *getenv(const char *name);</code>	读取环境变量 <code>name</code> 的当前值。如 <code>getenv("username")</code> 可以取得当前 Windows 用户名。
转换类	<code>double atof(const char *s);</code>	字符串转换为浮点数并返回。
	<code>int atoi(const char *s);</code>	字符串转换为整型数并返回。
	<code>char *itoa(int v, char *s, int radix);</code>	以 <code>radix</code> 进制将整数 <code>v</code> 转换到字符串 <code>s</code> 中并返回。

5. 时间函数库，`#include <time.h>`

类别	函数声明	功能
当前时间	<code>time_t time(time_t *p);</code> //typedef 定义 <code>time_t</code> 为 <code>long</code>	返回从 1970 年 1 月 1 日 0 点到当前时间的秒数，如果 <code>p</code> 不为空，将秒数存于 <code>*p</code> 中。
	<code>clock_t clock(void);</code> //typedef 定义 <code>clock_t</code> 为 <code>long</code>	返回从计算机上电启动到当前的时间数，以毫秒为单位（参考 <code>CLOCKS_PER_SEC</code> ，一般取值 1000）。

6. 控制台输入输出函数库，`#include <conio.h>`

类别	函数声明	功能
控制台输入输出	<code>int putch(int c);</code>	在控制台（屏幕）当前光标处显示字符 <code>c</code> ，同时光标右移，并返回字符 <code>c</code> 。
	<code>int getch(void);</code>	从控制台（键盘）上输入一个字符并返回。输入的字符不回显在屏幕上。 <code>getch</code> 可以用来读取上下左右等方向键和功能键，如依次返回 224, 72 表示向上键，依次返回 0, 94 表示按下 <code>Ctrl-F1</code> 等。
	<code>int kbhit(void);</code>	检查当前是否有键盘输入，若有则返回一个非 0 值，否则返回 0。
	<code>int ungetch(int c);</code>	将字符 <code>c</code> 退回到键盘输入缓冲区，成功返回字符 <code>c</code> ，否则返回 <code>EOF (-1)</code> 。

7. 字符类型函数库，`#include <ctype.h>`

类别	函数声明	功能
字符判断类	<code>int isdigit(int c);</code>	若 <code>c</code> 是数字字符（'0' ~ '9'）返回非 0 值，否则返回 0。
	<code>int islower(int c);</code>	若 <code>c</code> 是小写字母（'a' ~ 'z'）返回非 0 值，否则返回 0。
	<code>int isupper(int c);</code>	若 <code>c</code> 是大写字母（'A' ~ 'Z'）返回非 0 值，否则返回 0。
	<code>int isalpha(int c);</code>	若 <code>c</code> 是字母（'A' ~ 'Z', 'a' ~ 'z'）返回非 0 值，否则返回 0。

	<code>int isalnum(int c);</code>	若 c 是字母('A' ~ 'Z', 'a' ~ 'z')或数字字符('0' ~ '9') 返回非 0 值, 否则返回 0。
	<code>int isxdigit(int c);</code>	若 c 是十六进制数字字符('0' ~ '9', 'A' ~ 'F', 'a' ~ 'f') 返回非 0 值, 否则返回 0。
	<code>int isspace(int c);</code>	若 c 是空格(' '), 水平制表符('\t'), 换行符('\n'), 回车符('\r'), 垂直制表符('\v'), 翻页符('\f'), 返回非 0 值, 否则返回 0。
	<code>int iscntrl(int c);</code>	若 c 是 DEL 字符(0x7F)或普通控制字符(0x00 ~ 0x1F) 返回非 0 值, 否则返回 0。
大小写	<code>int tolower(int c);</code>	若 c 是大写字母('A' ~ 'Z') 返回相应的小写字母('a' ~ 'z'), 否则返回 c 本身。
	<code>int toupper(int c);</code>	若 c 是小写字母('a' ~ 'z') 返回相应的大写字母('A' ~ 'Z'), 否则返回 c 本身。

8. Windows 基础函数库, #include <winbase.h>

类别	函数声明	功能
延迟	<code>void Sleep(long t);</code>	Windows 调用函数(注意 Windows 函数首字母大写), 当前程序睡眠 t 毫秒后继续执行。