附表 A　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）。 如：char s[80];sprintf(s,"%d*%02d",15,2); 则 s 数组生成字符串"15*02"
	int **sscanf**(const char *s, const char *format, ...);	从字符缓冲区 s 上格式化输入数据，返回输入的数据个数。如：char s[80] = "15 02"; int a,b; sscanf(s,"%d%d",&a,&b);则变量 a 得到 15，变量 b 得到 2。

（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）。
	int **fgetc**(FILE *fp);	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。
	int **fputs**(const char *s,FILE *fp);	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。
	char * **fgets**(char *s,int n,FILE *fp);	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	int **fwrite**(const void *buff,int size,int n,FILE *fp);	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。
	int **fread**(void *buff,int size,int n,FILE *fp)	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	int **feof**(FILE *fp);	检查文件是否到达末尾，文件结束返回非 0 值，否则返回 0。
	long **ftell**(FILE *fp);	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。
	int **fseek**(FILE *fp,long offset,int from);	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。
	void **rewind**(FILE *fp);	重置文件流当前位置，回到文件开头。
其他	int **fflush**(FILE *fp);	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。
	int **ferror**(FILE *fp);	文件操作失败时，返回文件错误号，返回 0 表示没有错误。
	void **clearerr**(FILE *fp);	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	double **sqrt**(double x);	计算平方根
	double **exp**(double x);	指数函数，即 e^x
	double **pow**(double x, double y);	幂函数，即 x^y
	double **log**(double x);	对数函数 ln(x)，以 e 为底
	double **log10**(double x);	对数函数 $\log_{10}(x)$，以 10 为底
三角类	double **sin**(double x);	正弦函数
	double **cos**(double x);	余弦函数
	double **tan**(double x);	正切函数
	double **asin**(double x);	反正弦函数
	double **acos**(double x);	反余弦函数
	double **atan**(double x);	反正切函数
	double **sinh**(double x);	双曲正弦函数
	double **cosh**(double x);	双曲余弦函数
	double **tanh**(double x);	双曲正切函数
其	double **fabs**(double x);	返回浮点数的绝对值

他	double **floor**(double x);	向下取整，返回≤x 的最大整数。如：floor(7.9)的结果为 7.0，floor(7)的结果为 7.0。
	double **ceil**(double x);	向上取整，返回≥x 的最小整数。如：ceil(7.1)的结果为 8.0，ceil(8)的结果为 8.0。
	double **fmod**(double x, double y);	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	void **srand**(unsigned seed);	设置伪随机数生成种子，种子不同，随后调用 rand 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 srand(time(NULL))；
	int **rand**(void);	返回下一个伪随机数，伪随机数取值 0 至 RAND_MAX（32767）之间（含）。
系统类	void **exit**(int ret);	结束当前程序，并返回码 ret。exit 函数可以在任何函数中调用，在 main 函数中的作用等同于 return(ret)。
	int **system**(const char *command);	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 system("cls");可以清除程序运行窗口的屏幕。
	char ***getenv**(const char *name);	读取环境变量 name 的当前值。如 getenv("username")可以取得当前 Windows 用户名。
转换类	double **atof**(const char *s);	字符串转换为浮点数并返回。
	int **atoi**(const char *s);	字符串转换为整型数并返回。
	char ***itoa**(int v,char *s,int radix);	以 radix 进制将整数 v 转换到字符串 s 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	time_t **time**(time_t *p); //typedef 定义 time_t 为 long	返回从 1970 年 1 月 1 日 0 点到当前时间的秒数，如果 p 不为空，将秒数存于*p 中。 如：time_t now = time(NULL); //当前时间戳
	clock_t **clock**(void); //typedef 定义 clock_t 为 long	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 CLOCKS_PER_SEC，一般取值 1000）。

时间类型	日期时间结构体，struct tm {
	int tm_sec; //秒，取值[0,59]
	int tm_min; //分，取值[0,59]
	int tm_hour; //时，取值[0,23]
	int tm_mday; //日期，当月，取值[1,31]
	int tm_mon; //月份，取值 0 代表一月
	int tm_year; //年份，取 0 代表 1900
	int tm_wday; //星期，取 0 代表星期天
	int tm_yday; //当年天数
	int tm_isdst; }; //夏令时标识
		包含：年、月、日、时、分、秒、星期等信息。
		tm_year 代表年份，实际年份减去 1900
		tm_mon 代表月份，取值 0 至 11，0 代表一月份
		tm_mday 为当月开数，取值 1 至 31 日
		tm_yday 为当年 1 月 1 日开始的天数，取值[0,365]，取值 0 代表 1 月 1 日
		tm_isdst 为夏令时标识，夏令时为正，不实行夏令时为 0

时间转换	struct tm *localtime(time_t *);	从长整数的时间戳（秒数）转换为年月时时分秒形式，返回结构体指针。如取得当前年月日信息。 如：struct tm *t=localtime(&now);
	time_t mktime(struct tm *);	从年月时时分秒结构体形式转换为时间戳（秒数）
	int strftime(char *str, int n, char *format, struct tm *t);	转换为指定格式串，如：char str[80]; strftime(str, 80, "%Y 年%m 月%d 日 %H 时%M 分%S 秒", t);
	double difftime(time_t, time_t);	取得 2 个时间结构体之间的时间差，单位为秒。

6. 控制台输入输出函数库，#include <conio.h>

类别	函 数 声 明	功　　能
控制台输入出	int **putch**(int c);	在控制台（屏幕）当前光标处显示字符 c，同时光标右移，并返回字符 c。
	int **getch**(void);	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。getch 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 Ctrl-F1 等。
	int **kbhit**(void);	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。
	int **ungetch**(int c);	将字符 c 退回到键盘输入缓冲区，成功返回字符 c，否则返回 EOF（-1）。

7. 字符类型函数库，#include <ctype.h>

类别	函 数 声 明	功　　能
字符判断类	int **isdigit**(int c);	若 c 是数字字符('0'～'9')返回非 0 值,否则返回 0。
	int **islower**(int c);	若 c 是小写字母（'a'～'z'）返回非 0 值，否则返回 0。
	int **isupper**(int c);	若 c 是大写字母（'A'～'Z'）返回非 0 值，否则返回 0。
	int **isalpha**(int c);	若 c 是字母（'A'～'Z','a'～'z'）返回非 0 值，否则返回 0。
	int **isalnum**(int c);	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。
	int **isxdigit**(int c);	若 c 是十六进制数字符（'0'～'9','A'-'F','a'-'f'）返回非 0 值，否则返回 0。
	int **isspace**(int c);	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。
	int **iscntrl**(int c);	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	int **tolower**(int c);	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。
	int **toupper**(int c);	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	void **Sleep**(long t);	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。

附表 B　运算符优先级与结合性

优先级	运算符	名称或含义	使用形式	结合方向	说明
1	[]	数组下标	数组名[常量表达式]	左到右	
	()	圆括号	(表达式) 函数名(形参表)		
	.	成员选择（对象）	对象.成员名		
	->	成员选择（指针）	对象指针->成员名		
	++	自增运算符	变量名++		单目运算符
	--	自减运算符	变量名--		单目运算符
2	-	负号运算符	-常量	右到左	单目运算符
	(类型)	强制类型转换	(数据类型)表达式		
	++	自增运算符	++变量名		单目运算符
	--	自减运算符	--变量名		单目运算符
	*	取值运算符	*指针变量		单目运算符
	&	取地址运算符	&变量名		单目运算符
	!	逻辑非运算符	!表达式		单目运算符
	~	按位取反运算符	~表达式		单目运算符
	sizeof	长度运算符	sizeof(表达式)		
3	*	乘	表达式*表达式		双目运算符
	/	除	表达式/表达式		双目运算符
	%	余数（取模）	整型表达式/整型表达式		双目运算符
4	+	加	表达式+表达式	左到右	双目运算符
	-	减	表达式-表达式		双目运算符
5	<<	左移	变量<<表达式	左到右	双目运算符
	>>	右移	变量>>表达式		双目运算符
6	>	大于	表达式>表达式	左到右	双目运算符
	>=	大于等于	表达式>=表达式		双目运算符
	<	小于	表达式<表达式		双目运算符
	<=	小于等于	表达式<=表达式		双目运算符
7	==	等于	表达式==表达式	左到右	双目运算符
	!=	不等于	表达式!= 表达式		双目运算符

优先级	运算符	名称或含义	使用形式	结合方向	说明
8	&	按位与	表达式&表达式	左到右	双目运算符
9	^	按位异或	表达式^表达式	左到右	双目运算符
10	|	按位或	表达式|表达式	左到右	双目运算符
11	&&	逻辑与	表达式&&表达式	左到右	双目运算符
12	||	逻辑或	表达式||表达式	左到右	双目运算符
13	?:	条件运算符	表达式1? 表达式2: 表达式3	右到左	三目运算符
14	=	赋值运算符	变量=表达式	右到左	
	/=	除后赋值	变量/=表达式		
	*=	乘后赋值	变量*=表达式		
	%=	取模后赋值	变量%=表达式		
	+=	加后赋值	变量+=表达式		
	-=	减后赋值	变量-=表达式		
	<<=	左移后赋值	变量<<=表达式		
	>>=	右移后赋值	变量>>=表达式		
	&=	按位与后赋值	变量&=表达式		
	^=	按位异或后赋值	变量^=表达式		
	|=	按位或后赋值	变量|=表达式		
15	,	逗号运算符	表达式,表达式,…	左到右	从左向右顺序运算

附表 C　ASCII 码表

编码-十进制	编码-十六进制	转义字符 /字符功能		编码-十进制	编码-十六进制	转义字符 /字符功能
0	00	空字符，‘\0’ 字符串结束符		16	10	Ctrl-P（^P） 协议 DLE 数据链路转义
1	01	Ctrl-A（^A） 协议 SOH 标题开始		17	11	Ctrl-Q（^Q） 协议 DC1 设备控制 1
2	02	Ctrl-B（^B） 协议 STX 正文开始		18	12	Ctrl-R（^R） 协议 DC2 设备控制 2
3	03	Ctrl-C（^C） 中断输入 协议 ETX 正文结束		19	13	Ctrl-S（^S） 暂停滚动显示 协议 DC3 设备控制 3
4	04	Ctrl-D（^D） 协议 EOT 传输结束		20	14	Ctrl-T（^T） 协议 DC4 设备控制 4
5	05	Ctrl-E（^E） 协议 ENQ 请求		21	15	Ctrl-U（^U） 协议 NAK 拒绝接收
6	06	Ctrl-F（^F） 协议 ACK 收到通知		22	16	Ctrl-V（^V） 协议 SYN 同步空闲
7	07	Ctrl-G（^G），‘\a’ 响铃（发嘟声）		23	17	Ctrl-W（^W） 协议 ETB 结束传输块
8	08	Ctrl-H（^H），‘\b’ 退格键 删除光标前一个字符		24	18	Ctrl-X（^X） 协议 CAN 取消
9	09	Ctrl-I（^I），‘\t’ TAB 键 水平制表符		25	19	Ctrl-Y（^Y） 协议 EM 媒介结束
10	0A	Ctrl-J（^J），‘\n’ 换行符 光标跳到下一行第一列		26	1A	Ctrl-Z（^Z） 结束输入流 协议 SUB 代替
11	0B	Ctrl-K（^K），‘\v’ 垂直制表符		27	1B	ESC 键 协议 ESC 换码(溢出)
12	0C	Ctrl-L（^L），‘\f’ 换页符		28	1C	协议 FS 文件分隔符
13	0D	Ctrl-M（^M），‘\r’ Enter 回车键 光标回到同一行第一列		29	1D	协议 GS 分组符
14	0E	Ctrl-N（^N） 协议 SO 不用切换		30	1E	协议 RS 记录分隔符
15	0F	Ctrl-O（^O） 协议 SI 启用切换		31	1F	协议 US 单元分隔符

编码-十进制	编码-十六进制	转义字符/字符功能		编码-十进制	编码-十六进制	转义字符/字符功能
32	20	空格，‘ ‘		64	40	@，AT 符号
33	21	!，感叹号		65	41	大写字母 A
34	22	"，双引号，‘\"‘		66	42	大写字母 B
35	23	#，井号		67	43	大写字母 C
36	24	美元符		68	44	大写字母 D
37	25	$，百分号		69	45	大写字母 E
38	26	&，And 字符与运算符/取址运算符		70	46	大写字母 F
39	27	'，单引号，‘\‘		71	47	大写字母 G
40	28	(，左(小/圆)括号/开括号		72	48	大写字母 H
41	29	)，右(小/圆)括号/闭括号		73	49	大写字母 I
42	2A	*，乘号/星号/取值运算符		74	4A	大写字母 J
43	2B	+，加号		75	4B	大写字母 K
44	2C	,，逗号		76	4C	大写字母 L
45	2D	-，减号/破折号		77	4D	大写字母 M
46	2E	.，小数点/句号		78	4E	大写字母 N
47	2F	/，除号/斜杠		79	4F	大写字母 O
48	30	数字 0		80	50	大写字母 P
49	31	数字 1		81	51	大写字母 Q
50	32	数字 2		82	52	大写字母 R
51	33	数字 3		83	53	大写字母 S
52	34	数字 4		84	54	大写字母 T
53	35	数字 5		85	55	大写字母 U
54	36	数字 6		86	56	大写字母 V
55	37	数字 7		87	57	大写字母 W
56	38	数字 8		88	58	大写字母 X
57	39	数字 9		89	59	大写字母 Y
58	3A	:，冒号		90	5A	大写字母 Z
59	3B	;，分号		91	5B	[，左中括号/开方括号
60	3C	<，小于		92	5C	\，反斜杠，‘\\‘
61	3D	=，等号		93	5D	]，右中括号/闭方括号
62	3E	>，大于		94	5E	^，异或运算符/脱字符
63	3F	?，问号		95	5F	_，下划线

编码-十进制	编码-十六进制	转义字符/字符功能		编码-十进制	编码-十六进制	转义字符/字符功能
96	60	`，反单引号/重音符		112	70	小写字母 p
97	61	小写字母 a		113	71	小写字母 q
98	62	小写字母 b		114	72	小写字母 r
99	63	小写字母 c		115	73	小写字母 s
100	64	小写字母 d		116	74	小写字母 t
101	65	小写字母 e		117	75	小写字母 u
102	66	小写字母 f		118	76	小写字母 v
103	67	小写字母 g		119	77	小写字母 w
104	68	小写字母 h		120	78	小写字母 x
105	69	小写字母 i		121	79	小写字母 y
106	6A	小写字母 j		122	7A	小写字母 z
107	6B	小写字母 k		123	7B	{，左大括号/开花括号
108	6C	小写字母 l		124	7C	|，或运算符/垂线/竖线
109	6D	小写字母 m		125	7D	}，右大括号/闭花括号
110	6E	小写字母 n		126	7E	~，波浪号
111	6F	小写字母 o		127	7F	DEL，删除

注：

（1）本表列出 128 个西文字符，ASCII 编码 0 至 127，对应十六进制编码从 00 至 7F。

（2）ASCII 码表中，数字字符 0 至 9 连续编码（编码 48~57），大写字母 A 至 Z 连续编码（编码 65~90），小写字母 a 至 z 也是连续编码（编码 97~122），大小写字母之间编码相差 32。

（3）编码 0 至 31 为控制字符，C 语言学习中，主要使用到的有：空字符'\0'、响铃'\a'、退格'\b'、换行'\n'、回车'\r'等。

① 在命令行环境中，键入 Ctrl-A 对应 ASCII 编码 1，回显^A。编码 1 至 26，对应 Ctrl-A 至 Ctrl-Z，回显^A 至^Z。键入 Ctrl-C 表示中断当前程序，Ctrl-H 对应 BackSpace 退格键，Ctrl-I 对应 TAB 键，Ctrl-M 对应回车键。

② 编码 1 至 31 等字符主要用于通讯协议，编码 1 表示 SOH（Start Of Head，标题开始），初学者可以不必理会。