

本附录描述了标准C支持的库函数^①。使用此附录时，请记住下列要点。

- 为了简洁清楚，这里删除了一些细节。如果想看全部内容，请参考标准。本书的其他地方已经对一些函数（特别是printf函数、scanf函数以及它们的变异函数）进行了详细介绍，所以这里只对这类函数做简短的描述。为了获得关于某个函数更详细的信息（包括如何使用这个函数的示例），请见函数描述右下角用楷体列出的节号。
- 每个函数描述结尾都有其他与之相关函数的列表。**相似函数**非常接近于正在描述的函数。**相关函数**经常会和在描述的函数联合使用。（例如，calloc函数和realloc函数与malloc函数“类似”，而free函数则与malloc函数“相关”。）**也可参见**的函数和在描述的函数没有紧密联系，但是却可能有影响。
- 如果把函数行为的某些方面描述为由**实现定义的**，那么这就意味着此函数依赖于C库的实现方式。函数将始终行为一致，但是结果却可能会由于系统的不同而千差万别。（换句话说，请参考手册了解可能发生的问题。）另一方面，**未定义**的行为是一个不好的消息：不但函数的行为可能会因系统不同而不同，而且程序也可能会行为异常甚至崩溃。
- <math.h>中许多函数的描述提到了**定义域错误**和**取值范围错误**。在本附录的末尾对这两种错误进行了定义。
- 下列库函数的行为是会受到当前地区影响的：
 - 字符处理函数（除了isdigit函数和isxdigit函数）。
 - 格式化输入/输出函数。
 - 多字节字符和字符串函数。
 - 字符串转换函数。
 - Strcoll函数、strftime函数和strxfrm函数。例如，isalpha函数实际上检测字符是否在a到z之间或者在A到Z之间。在某些区域内也把其他字符看成是字母次序的。本附录描述了在“C”（默认的）地区内库函数的行为。
- 一些函数实际上是宏。然而，这些宏的用法和函数完全一样，所以这里不对它们区别对待。

abort	异常终止程序	<stdlib.h>
	void abort(void);	
	产生SIGABRT信号。如果无法捕获信号（或者如果信号处理函数返回），那么程序会异常终止，并且返回由实现定义的代码来说明不成功的终止。是否清洗输出缓冲区，是否关闭打开的流，以及是否移除临时文件都是由实现定义的。	
相似函数	exit函数、raise函数	
相关函数	assert函数、signal函数	
也可参见	atexit函数	26.2节

abs	整数的绝对值	<stdlib.h>
	int abs(int j);	
返回	整数j的绝对值。如果不能表示j的绝对值，那么函数的行为是未定义的。	

① 这些材料经ANSI许可可改编自American National Standards Institute ANSI/ISO 9899©1990。这个标准的副本可从ANSI购买（ANSI, 11 West 42nd Street, New York, NY 10036）。

相似函数	fabs函数、labs函数	26.2节
acos	反余弦 <code>double acos(double x);</code>	<math.h>
返回	x的反余弦值。返回值的范围在0到 π 之间。如果x的值不在-1到+1之间，那么就会发生定义域错误。	
相关函数	asin函数、atan函数、atan2函数、cos函数、sin函数、tan函数	23.3节
asctime	把日期和时间转换成ASCII码 <code>char *asctime(const struct tm *timeptr);</code>	<time.h>
返回	指向以空字符结尾的字符串的指针，其格式如下所示： Mon Jul 15 12:30:45 1996\n 此格式的构造来源于timeptr指向的结构中的分解时间。	
相似函数	ctime函数、strftime函数	
相关函数	diffime函数、gmtime函数、localtime函数、mktime函数、time函数	26.3节
asin	反正弦 <code>double asin(double x);</code>	<math.h>
返回	x的反正弦值。返回值的范围在 $-\pi/2$ 到 $\pi/2$ 之间。如果x的值不在-1到+1之间，那么就会发生定义域错误。	
相关函数	acos函数、atan函数、atan2函数、cos函数、sin函数、tan函数	23.3节
assert	诊断表达式的真值 <code>void assert(int expression);</code> 如果expression的值非零，那么assert函数什么也不做。如果expression的值为零，那么assert函数向stderr写信息（说明expression的文本，含有assert函数的源文件名，以及assert函数的行数），然后通过调用abort函数终止程序。为了使assert函数无效，要在包含<assert.h>之前定义宏NDEBUG。	<assert.h>
相关函数	abort函数	24.1节
atan	反正切 <code>double atan(double x);</code>	<math.h>
返回	x的反正切值。返回值的范围在 $-\pi/2$ 到 $\pi/2$ 之间。	
相似函数	atan2函数	
相关函数	acos函数、asin函数、cos函数、sin函数、tan函数	23.3节
atan2	商的反正切 <code>double atan2(double y, double x);</code>	<math.h>
返回	y/x的反正切值。返回值的范围在 $-\pi$ 到 π 之间。如果x和y的值都为零，那么就会发生定义域错误。	
相似函数	atan函数	
相关函数	acos函数、asin函数、cos函数、sin函数、tan函数	23.3节
atexit	在程序退出处注册要调用的函数 <code>int atexit(void (*func)(void));</code> 注册由func指向的函数作为终止函数。如果程序正常终止（通过return或exit，而不是abort），那么将调用函数。可以重复调用atexit函数来注册多个终止函数。最后一个注册的函数将是在终止前第一个被调用的函数。	<stdlib.h>
返回	如果成功，返回零。如果不成功，则返回非零（达到由实现定义的限制）。	
相关函数	exit函数	
也可参见	abort函数	26.2节
atof	把字符串转换成浮点数 <code>double atof(const char *nptr);</code>	<stdlib.h>

	<code>double atof(const char *nptr);</code>	
返回	对应字符串最长初始部分的double型值，此字符串是由nptr指向的，且字符串最长初始部分具有浮点数的格式。如果无法表示此数，那么函数的行为将是未定义的。	
相似函数	strtod函数	
相关函数	atoi函数、atol函数	
也可参见	strtol函数、strtoul函数	26.2节
atoi	把字符串转换成整数	<stdlib.h>
	<code>int atoi(const char *nptr);</code>	
返回	对应字符串最长初始部分的整数，此字符串是由nptr指向的，且字符串最长初始部分具有整数的格式。如果无法表示此数，那么函数的行为将是未定义的。	
相似函数	atol函数、strtol函数、strtoul函数	
相关函数	atof函数	
也可参见	strtod函数	26.2节
atol	把字符串转换成长整数	<stdlib.h>
	<code>long int atol(const char *nptr);</code>	
返回	对应字符串最长初始部分的长整数，此字符串是由nptr指向的，且字符串最长初始部分具有整数的格式。如果无法表示此数，那么函数的行为将是未定义的。	
相似函数	atoi函数、strtol函数、strtoul函数	
相关函数	atof函数	
也可参见	strtod函数	26.2节
bsearch	二分检索	<stdlib.h>
	<code>void *bsearch(const void *key, const void *base, size_t memb, size_t size, int (*compar)(const void *, const void *));</code>	
	在有序数组中搜索由key指向的值。其中，数组存储在base地址上，且此数组有nmemb个元素，每个元素大小为size个字节。compar指向“比较函数”。换句话说当传递指向关键字的指针和数组元素时，比较函数必须返回负整数、零或正整数，这主要依赖于关键字是小于、等于还是大于数组元素。	
返回	指向数组元素的指针，此数组元素是用来测试是否等于关键字的。如果没有找到关键字，那么返回空指针。	
相关函数	qsort函数	26.2节
calloc	分配并清除内存块	<stdlib.h>
	<code>void *calloc(size_t nmemb, size_t size);</code>	
	为带有nmemb个元素的数组分配内存块，其中每个数组元素占size个字节。通过设置所有位为零来清除内存块。	
返回	指向内存块开始处的指针。如果不能分配所要求大小的内存块，那么返回空指针。	
相似函数	malloc函数、realloc函数	
相关函数	free函数	17.3节
ceil	上整数	<math.h>
	<code>double ceil(double x);</code>	
返回	大于或等于x的最小整数。	
相似函数	floor函数	23.3节
clearerr	清除流错误	<stdio.h>
	<code>void clearerr(FILE *stream);</code>	
	为stream指向的流清除文件尾指示器和错误指示器。	
相关函数	feof函数、ferror函数、rewind函数	22.3节

clock	处理器时钟	<time.h>
	clock_t clock(void);	
返回	从程序开始执行起所经过的处理器时间（按照“时钟嘀嗒”来衡量的）。（用CLOCKS_PER_SEC除以时间来转换成秒。）如果时间无效或者无法表示，那么返回(clock_t)-1。	
相似函数	time函数	
也可参见	difftime函数	26.3节
cos	余弦	<math.h>
	double cos(double x);	
返回	x的余弦值（按照弧度衡量的）。	
也可参见	acos函数、asin函数、atan函数、atan2函数、sin函数、tan函数	23.3节
cosh	双曲余弦	<math.h>
	double cosh(double x);	
返回	x的双曲余弦值。如果x的数过大，那么可能会发生取值范围错误。	
相关函数	sinh函数、tanh函数	
也可参见	acos函数、asin函数、atan函数、atan2函数、cos函数、sin函数、tan函数	23.3节
ctime	把日期和时间转换成字符串	<time.h>
	char *ctime(const time_t *timer);	
返回	指向字符串的指针，此字符串描述了本地时间，此时间等价于timer指向的日历时间。等价于asctime(localtime(timer))。	
相似函数	asctime函数、strftime函数	
相关函数	difftime函数、gmtime函数、localtime函数、mktime函数、time函数	26.3节
difftime	时间差	<time.h>
	double difftime(time_t time1, time_t time0);	
返回	time0（较早的时间）和time1之间的差值，此值按秒来衡量。	
相关函数	asctime函数、ctime函数、gmtime函数、localtime函数、mktime函数、strftime函数、time函数	
也可参见	clock函数	26.3节
div	整数除法	<stdlib.h>
	div_t div(int numer, int denom);	
返回	含有quot（numer除以denom时的商）和rem（余数）的结构。如果无法表示结果，那么函数的行为是未定义的。	
相似函数	ldiv函数	26.2节
exit	退出程序	<stdlib.h>
	void exit(int status);	
	调用所有用atexit函数注册的函数，清洗全部输出缓冲区，关闭所有打开的流，移除任何由tmpfile产生的文件，并终止程序。status的值说明程序是否正常终止。status唯一可移植的值是0和EXIT_SUCCESS（两者都说明成功终止）以及EXIT_FAILURE（不成功的终止）。status的其他值都是由实现定义的。	
相似函数	abort函数	
相关函数	atexit函数	9.5节、26.2节
exp	指数	<math.h>
	double exp(double x);	
返回	e的x次幂的值（即e ^x ）。如果x的数过大，那么可能会发生取值范围错误。	
相似函数	pow函数	
相关函数	log函数	

也可参见	log10函数	23.3节
fabs	浮点数的绝对值 double fabs(double x); 返回 x的绝对值。 相似函数 abs函数、labs函数	<math.h> 23.3节
fclose	关闭文件 int fclose(FILE *stream); 关闭由stream指向的流。清洗保留在流缓冲区内的任何未写的输出。如果是自动分配，那么就释放缓冲区。 返回 如果成功，就返回零。如果检测到错误，就返回EOF。 相关函数 fopen函数、freopen函数 也可参见 fflush函数	<stdio.h> 22.2节
feof	检测文件末尾 int feof(FILE *stream); 返回 如果为stream指向的流设置了文件尾指示器，那么返回非零值。否则返回零。 相似函数 ferror函数 相关函数 clearerr函数、fseek函数、rewind函数	<stdio.h> 22.3节
ferror	检测文件错误 int ferror(FILE *stream); 返回 如果为stream指向的流设置了文件错误指示器，那么返回非零值。否则返回零。 相似函数 feof函数 相关函数 clearerr函数、rewind函数	<stdio.h> 22.3节
fflush	清洗文件缓冲区 int fflush(FILE *stream); 把任何未写入的数据写到和stream相关的缓冲区中，其中stream指向用于输出或更新的已打开的流。如果stream是空指针，那么fflush函数清洗存储在缓冲区中的所有未写入的流。 返回 如果成功就返回零。如果检测到错误，就返回EOF。 也可参见 fclose函数、setbuf函数、setvbuf函数	<stdio.h> 22.2节
fgetc	从文件中读取字符 int fgetc(FILE *stream); 从stream指向的流中读取字符。 返回 读到的字符。如果fgetc函数遇到流的末尾，则设置流的文件尾指示器并且返回EOF。如果读取发生错误，fgetc函数设置流的错误指示器并且返回EOF。 相似函数 getc函数、getchar函数 相关函数 fputc函数、putc函数、ungetc函数 也可参见 putchar函数	<stdio.h> 22.4节
fgetpos	获得文件位置 int fgetpos(FILE *stream, fpos_t *pos); 把stream指向的流的当前位置存储到pos指向的对象中。 返回 如果成功就返回零。如果调用失败，则返回非零值，并且把由实现定义的错误码存储到errno中。 相似函数 ftell函数 相关函数 fsetpos函数 也可参见 fseek函数、rewind函数	<stdio.h> 22.7节
fgets	从文件中读取字符串 char *fgets(char *s, int n, FILE *stream);	<stdio.h>

607

608

	从stream指向的流中读取字符，并且把读入的字符存储到s指向的数组中。遇到第一个换行符已经读取了n-1个字符，或到了文件末尾时，读取操作都会停止。fgets函数会在字符串后添加一个空字符。	
返回	s（指向数组的指针，此数组存储着输入）。如果读取操作错误或fgets函数在存储任何字符之前遇到了流的末尾，都会返回空指针。	
相似函数	gets函数	
相关函数	fputs函数	
也可参见	puts函数	22.5节
floor	向下取整	<math.h>
	double floor(double x);	
返回	小于或等于x的最大整数。	
相似函数	ceil函数	23.3节
fmod	浮点模数	<math.h>
	double fmod(double x, double y);	
返回	x除以y的余数。如果y为零，是发生定义域错误还是fmod函数返回零是由实现定义的。	
也可参见	div函数、ldiv函数	23.3节
fopen	打开文件	<stdio.h>
	FILE *fopen(const char *filename, const char *mode); 打开文件以及和它相关的流，文件名是由filename指向的。mode说明文件打开的方式。为流清除错误指示器和文件尾指示器。	
返回	文件指针。在执行下一次关于文件的操作时会用到此指针。如果无法打开文件则返回空指针。	
相似函数	freopen函数	
相关函数	fclose函数、setbuf函数、setvbuf函数	22.2节
fprintf	格式化写文件	<stdio.h>
	int fprintf(FILE *stream, const char *format, ...); 向stream指向的流写输出。format指向的字符串说明了后续参数显示的格式。	
返回	写入的字符数量。如果发生错误就返回负值。	
相似函数	printf函数、sprintf函数、vfprintf函数、vprintf函数、vsprintf函数	
相关函数	fscanf函数	
也可参见	scanf函数、sscanf函数	22.3节
fputc	向文件写字符	<stdio.h>
	int fputc(int c, FILE *stream); 把字符c写到stream指向的流中。	
返回	c（写入的字符）。如果写发生错误，fputc函数会为stream设置错误指示器，并且返回EOF。	
相似函数	putc函数、putchar函数	
相关函数	fgetc函数、getc函数	
也可参见	getchar函数	22.4节
fputs	向文件写字符串	<stdio.h>
	int fputs(const char *s, FILE *stream); 把s指向的字符串写到stream指向的流中。	
返回	如果成功，返回非负值。如果写发生错误，则返回EOF。	
相似函数	puts函数	
相关函数	fgets函数	
也可参见	gets函数	22.5节
fread	从文件读块	<stdio.h>
	size_t fread(void *ptr, size_t size, size_t nmemb, FILE *stream);	

610	返回	<p>试着从stream指向的流中读取nmemb个元素，每个元素大小为size个字节，并且把读入的元素存储到ptr指向的数组中。</p> <p>实际读入的元素（不是字符）数量。如果fread遇到文件末尾或检测到读取错误，那么此数将会小于nmemb。如果nmemb或size为零，则返回值为零。</p>	22.6节
相关函数	fwrite函数		
free	释放内存块	<stdlib.h>	
<pre>void free (void *ptr);</pre> <p>释放地址为ptr的内存块（除非ptr为空指针时调用无效）。块必须通过calloc函数、malloc函数或realloc函数进行分配。</p>			
相关函数	calloc函数、malloc函数、realloc函数		17.4节
freopen	重新打开文件	<stdio.h>	
<pre>FILE *freopen(const char *filename, const char *mode, FILE *stream);</pre>			
在freopen函数关闭和stream相关的文件后，打开名为filename且与stream相关的文件。Mode参数具有和fopen函数调用中相同的含义。			
返回	如果操作成功，返回stream的值。如果无法打开文件则返回空指针。		
相似函数	fopen函数		
相关函数	fclose函数、setbuf函数、setvbuf函数		22.2节
frexp	分解成小数和指数	<math.h>	
<pre>double frexp(double value, int *exp);</pre> <p>按照下列形式把value分解成小数部分<i>f</i>和指数部分<i>n</i>： $value = f \times 2^n$ 其中<i>f</i>是规范化的，因此$0.5 \leq f < 1$或者<i>f</i>=0。把<i>n</i>存储在exp指向的整数中。</p>			
返回	<i>f</i> ，即value的小数部分。		
相关函数	ldexp函数		
也可参见	modf函数		23.3节
fscanf	格式化读文件	<stdio.h>	
611	<pre>int fscanf(FILE *stream, const char *format, ...);</pre> <p>向stream指向的流读入任意数量的数据项。format指向的字符串说明了读入项的格式。跟在format后边的参数指向数据项存储的位置。</p>		
返回	成功读入并且存储的数据项数量。如果发生错误或在可以读数据项前到达了文件末尾，那么就返回EOF。		
相似函数	scanf函数、sscanf函数		
相关函数	fprintf函数、vfprintf函数		
也可参见	printf函数、sprintf函数、vprintf函数、vsprintf函数		22.3节
fseek	文件查找	<stdio.h>	
<pre>int fseek(FILE *stream, long int offset, int whence);</pre> <p>为stream指向的流改变文件位置指示器。如果whence是SEEK_SET，那么新位置是在文件开始处加上offset个字节。如果whence是SEEK_CUR，那么新位置是在当前位置加上offset个字节。如果whence是SEEK_END，那么新位置是在文件末尾加上offset个字节。对于文本流而言，offset必须是零，或者whence必须是SEEK_SET并且offset的值是由前一次的ftell函数调用获得的。而对于二进制流来说，fseek函数不可以支持whence是SEEK_END的调用。</p>			
返回	如果操作成功就返回零。否则返回非零值。		
相似函数	fsetpos函数、rewind函数		
相关函数	ftell函数		
也可参见	fgetpos函数		22.7节

fsetpos	设置文件位置	<stdio.h>
	int fsetpos(FILE *stream, const fpos_t *pos); 根据pos（前一次fgetpos函数调用获得的）指向的值为stream指向的流设置文件位置指示器。	
返回	如果成功就返回零。如果调用失败，返回非零值，并且把由实现定义的错误码存储在errno中。	
相似函数	fseek函数、rewind函数	
相关函数	fgetpos函数	
也可参见	ftell函数	22.7节
ftell	确定文件位置	<stdio.h>
	long int ftell(FILE *stream);	
返回	返回stream指向的流的当前文件位置指示器。如果调用失败，返回-1L，并且把由实现定义的错误码存储在errno中。	
相似函数	fgetpos函数	
相关函数	fseek函数	
也可参见	fsetpos函数、rewind函数	22.7节
fwrite	向文件写块	<stdio.h>
	size_t fwrite(const void *ptr, size_t size, size_t nmemb, FILE *stream);	
	从ptr指向的数组中写nmemb个元素到stream指向的流中，且每个元素大小为size个字节。	
返回	实际写入的元素（不是字符）的数量。如果fwrite函数检测到写错误，则这个数将会小于nmemb。	
相关函数	fread函数	22.6节
getc	从文件读入字符	<stdio.h>
	int getc(FILE *stream); 从stream指向的流中读入一个字符。注意：getc函数通常是作为宏来实现的。它可能计算stream不只一次。	
返回	读入的字符。如果getc函数遇到流的末尾，那么它会设置流的文件尾指示器并且返回EOF。如果读取发生错误，那么getc函数设置流的错误指示器并且返回EOF。	
相似函数	fgetc函数、getchar函数	
相关函数	fputc函数、putc函数、ungetc函数	
也可参见	putchar函数	22.4节
getchar	读入字符	<stdio.h>
	int getchar(void); 从stdin流中读入一个字符。注意：getchar函数通常是作为宏来实现的。	
返回	读入的字符。如果getc函数遇到输入流的末尾，那么它会设置stdin流的文件尾指示器并且返回EOF。如果读取发生错误，那么getc函数设置stdin流的错误指示器并且返回EOF。	
相似函数	fgetc函数、getc函数	
相关函数	putchar函数、ungetc函数	
也可参见	fputc函数、putc函数	7.3节、22.4节
getenv	获取外部环境字符串	<stdlib.h>
	char *getenv(const char *name); 为了检查是否有任意字符串匹配name指向的字符串，搜索操作系统的外部环境列表。	
返回	与匹配名相关的字符串的指针。如果没有找到匹配则返回空指针。	
也可参见	system函数	26.2节
gets	读入字符串	<stdio.h>
	char *gets(char *s); 从stdin流中读入多个字符，并且把这些读入的字符存储到s指向的数组中。	

返回	s (即存储输入的数组的指针)。如果读取发生错误或gets函数在存储任何字符之前遇到流的末尾, 那么返回空指针。	
相似函数	fgets函数	
相关函数	puts函数	
也可参见	fputs函数	13.3节、22.5节
gmtime	转换成格林威治标准时间	<time.h>
	struct tm *gmtime(const time_t *timer);	
返回	指向结构的指针, 此结构包含的分解的UTC (协调世界时间—从前的格林威治时间) 值等价于timer指向的日历时间。如果UTC无效, 则返回空指针。	
相似函数	localtime函数	
相关函数	asctime函数、ctime函数、difftime函数、mktime函数、strftime函数、time函数	26.3节
isalnum	测试是字母或数字	<ctype.h>
	int isalnum(int c);	
返回	如果isalnum是字母或数字, 返回非零值; 否则返回零。(如果isalpha(c)或isdigit(c)为真, 则c是字母或数字。)	
相关函数	isalpha函数、isdigit函数	
也可参见	islower函数、isupper函数	23.4节
isalpha	测试字母	<ctype.h>
	int isalpha(int c);	
返回	如果isalnum是字母, 返回非零值; 否则返回零。(如果islower(c)或isupper(c)为真, 则c是字母。)	
相似函数	islower函数、isupper函数	
相关函数	isalnum函数	
也可参见	tolower函数、toupper函数	23.4节
iscntrl	测试控制字符	<ctype.h>
	int iscntrl(int c);	
返回	如果c是控制字符, 返回非零值; 否则返回零。	
相关函数	isgraph函数、isprint函数、isspace函数	23.4节
isdigit	测试数字	<ctype.h>
	int isdigit(int c);	
返回	如果c是数字, 返回非零值; 否则返回零。	
相似函数	isxdigit函数	
相关函数	isalnum函数	23.4节
isgraph	测试图形字符	<ctype.h>
	int isgraph(int c);	
返回	如果c是显示字符 (除了空格), 返回非零值; 否则返回零。	
相似函数	isprint函数	
相关函数	iscntrl函数、isspace函数	23.4节
islower	测试小写字母	<ctype.h>
	int islower(int c);	
返回	如果c是小写字母, 返回非零值; 否则返回零。	
相似函数	isalpha函数、isupper函数	
相关函数	tolower函数、toupper函数	
也可参见	isalnum函数	23.4节
isprint	测试显示字符	<ctype.h>

	<code>int isprint(int c);</code>	
返回	如果c是显示字符（包括空格），返回非零值；否则返回零。	
相似函数	<code>isgraph</code> 函数	
相关函数	<code>isctrnl</code> 函数、 <code>isspace</code> 函数	23.4节
ispunct	测试标点字符	<ctype.h>
	<code>int ispunct(int c);</code>	
返回	如果c是标点符号字符，返回非零值；否则返回零。除了空格、字母和数字字符以外，所有显示字符都可以看成是标点符号。	
也可参见	<code>isalnum</code> 函数、 <code>isgraph</code> 函数、 <code>isprint</code> 函数	23.4节
isspace	测试空白字符	<ctype.h>
	<code>int isspace(int c);</code>	
返回	如果c是空白字符，返回非零值；否则返回零。空白字符有空格（' '）、换页符（'\f'）、换行符（'\n'）、回车符（'\r'），横向制表符（'\t'）和纵向制表符（'\v'）。	
也可参见	<code>isctrnl</code> 函数、 <code>isgraph</code> 函数、 <code>isprint</code> 函数	23.4节
isupper	测试大写字母	<ctype.h>
	<code>int isupper(int c);</code>	
返回	如果c是大写字母，返回非零值；否则返回零。	
相似函数	<code>isalpha</code> 函数、 <code>islower</code> 函数	
也可参见	<code>tolower</code> 函数、 <code>toupper</code> 函数	23.4节
isxdigit	测试十六进制数字	<ctype.h>
	<code>int isxdigit(int c);</code>	
返回	如果c是十六进制数字（0-9、a-f、A-F），返回非零值；否则返回零。	
相似函数	<code>isdigit</code> 函数	23.4节
labs	长整数的绝对值	<stdlib.h>
	<code>longint labs(long int j);</code>	
返回	j的绝对值。如果不能表示j的绝对值，那么函数的行为是未定义的。	
相似函数	<code>abs</code> 函数、 <code>fabs</code> 函数	26.2节
ldexp	联合小数和指数	<math.h>
	<code>double ldexp(double x, int exp);</code>	
返回	$x \times 2^{\text{exp}}$ 的值。可能会发生取值范围错误。	
相关函数	<code>frexp</code> 函数	23.3节
ldiv	长整数除法	<stdlib.h>
	<code>ldiv_t ldiv(long int numer, long int denom);</code>	
返回	含有quot（numer除以denom的商）和rem（余数）的结构。如果无法表示结果，那么函数的行为是未定义的。	
相似函数	<code>div</code> 函数	26.2节
localeconv	获取区域转换	<locale.h>
	<code>struct lconv *localeconv(void);</code>	
返回	指向结构的指针，此结构含有当前区域信息。	
相关函数	<code>setlocale</code> 函数	25.1节
localtime	转换成区域时间	<time.h>
	<code>struct tm *localtime(const time_t *timer);</code>	
返回	指向结构的指针，此结构含有的分解时间等价于timer指向的日历时间。	
相似函数	<code>gmtime</code> 函数	
相关函数	<code>asctime</code> 函数、 <code>ctime</code> 函数、 <code>difftime</code> 函数、 <code>mktime</code> 函数、 <code>strftime</code> 函数、 <code>time</code> 函数	26.3节

615

616

617	log	自然对数 double log(double x); 返回 基数为e的x的对数(即lnx)。如果x是负数,会发生定义域错误;如果x是零,则会发生取值范围错误。 相似函数 log10函数 相关函数 exp函数 也可参见 pow函数	<math.h> 23.3节
	long10	常用对数 double log10(double x); 返回 基数为10的x的对数。如果x是负数,会发生定义域错误;如果x是零,则会发生取值范围错误。 相似函数 log函数 也可参见 exp函数、pow函数	<math.h> 23.3节
	longjmp	非区域跳转 void longjmp(jmp_buf env, int val); 恢复存储在env中的外部环境,并且从初始保存env的setjmp调用中返回。如果val非零,它将是setjmp的返回值;如果val为1,则setjmp返回1。 相关函数 setjmp函数 也可参见 signal函数	<setjmp.h> 24.4节
	malloc	分配内存块 void *malloc(size_t size); 分配size个字节的内存块。不清除内存块。 返回 指向内存块开始处的指针。如果无法分配要求尺寸的内存块,那么返回空指针。 相似函数 calloc函数、realloc函数 相关函数 free函数	<stdlib.h> 17.2节
	mblen	计算多字节字符的长度 int mblen(const char *s, size_t n); 如果s是空指针,则初始化移位状态。 返回 如果s是空指针,返回非零值还是零值依赖于多字节字符是否是依赖状态编码。如果s指向空字符则返回零;如果接下来n个或几个字节形成了一个有效的字符,那么返回s指向的多字节字符中的字节数量;否则返回-1。 相关函数 mbtowc函数、wctomb函数 也可参见 mbstowcs函数、setlocale函数、wcstombs函数	<stdlib.h> 25.2节
618	mbstowcs	把多字节字符串转换成宽字符串 size_t mbstowcs(wchar_t *pwcs, const char *s, size_t n); 把s指向的多字节字符序列转换为宽字符序列,并把不多于n个的编码存储到pwcs指向的数组中。如果遇到空字符则转换结束。空字符会被转换成为零值码。 返回 修改的数组元素的个数,无论如何也不包括终止码。如果遇到无效的多字节字符,则返回(size_t)-1。 相关函数 wctombs函数 也可参见 mblen函数、mbtowc函数、setlocale函数、wctomb函数	<stdlib.h> 25.2节
	mbtowc	把多字节字符转换成宽字符 int mbtowc(wchar_t *pwcs, const char *s, size_t n); 如果s是空指针,则初始化移位状态。如果s不是空指针,把s指向的多字节字符转换成宽字符码。最多将检查n个字节的字符。如果多字节字符有效,并且pwc不是空指	<stdlib.h>

	针，则把码存储到pwc指向的对象中。	
返回	如果s是空指针，则返回非零值还是零值依赖于多字节字符是否是依赖状态编码。如果s指向空字符，则返回零。如果接下来n个或几个字节形成了一个有效的字符，那么返回s指向的多字节字符中的字节数量。如果不是这样，则返回-1。	
相关函数	mblen函数、wctomb函数	
也可参见	mbstowcs函数、setlocale函数、wcstombs函数	25.2节
memchr	搜索内存块字符	<string.h>
	void *memchr(const void *s, int c, size_t n);	
返回	指向字符的指针，此字符是s所指向对象的前n个字符中第一个遇到的字符c。如果没有找到c，则返回空指针。	
相似函数	strchr函数	
也可参见	strpbrk函数、strrchr函数、strstr函数	23.5节
memcmp	比较内存块	<string.h>
	int memcmp(const void *s1, const void *s2, size_t n);	
返回	负整数、零还是正整数依赖于s1所指向对象的前n个字符是小于、等于还是大于s2所指向对象的前n个字符。	
相似函数	strcmp函数、strcoll函数、strncmp函数	23.5节
memcpy	复制内存块	<string.h>
	void *memcpy(void *s1, const void *s2, size_t n);	
	把s2所指向对象的n个字符复制到s1所指向的对象中。如果对象重叠，则不可能正确地工作。	
返回	s1（指向目的的指针）。	
相似函数	memmove函数、strcpy函数、strncpy函数	23.5节
memmove	复制内存块	<string.h>
	void *memmove(void *s1, const void *s2, size_t n);	
	把s2所指向对象的n个字符复制到s1所指向的对象中。如果对象重叠，即使memmove函数比memcpy函数速度慢，但是memmove函数还将正确地工作。	
返回	s1（指向目的的指针）。	
相似函数	memcpy函数、strcpy函数、strncpy函数	23.5节
memset	初始化内存块	<string.h>
	void *memset(void *s, int c, size_t n);	
	把c存储到s指向的内存块的前n个字符中。	
返回	s（指向内存块的指针）。	
相似函数	memcpy函数、memmove函数	23.5节
mktime	转换成日历时间	<time.h>
	time_t mktime(struct tm *timeptr);	
	把分解的区域时间（存储在由timeptr指向的结构中）转换成日历时间。结构的成员不要求一定在合法的取值范围内。而且，会忽略tm_wday（星期的天号）的值和tm_yday（年份的天号）的值。调整其他成员到正确的取值范围之内之后，mktime函数把值存储在tm_wday和tm_yday中。	
返回	日历时间对应timeptr指向的结构。如果无法表示日历时间，则返回(time_t)-1。	
相关函数	asctime函数、ctime函数、difftime函数、gmtime函数、localtime函数、strftime函数、time函数	26.3节
modf	分解成整数和小数部分	<math.h>
	double modf(double value, double *iptr);	
	把value分解成整数部分和小数部分。把整数部分存储到iptr指向的double型对象中。	
返回	value的小数部分。	

也可参见	frexp函数	23.3节
perror	显示错误信息 void perror(const char *s); 向Stderr流中写下列信息: 字符串: 出错信息 这里的字符串是s所指向的字符串。出错信息是由实现定义的,它与strerror (errno)函数调用返回的信息相匹配。	<stdio.h>
相关函数	strerror函数	24.2节
pow	幂 double pow(double x, double y); 返回 x的y次幂。发生定义域错误的情况有(1)当x是负数并且y的值不是整数时;或者(2)当x为零且y是小于或等于零,无法表示结果时。取值范围错误也是可能发生的。	<math.h>
相似函数	exp函数、sqrt函数	
也可参见	log函数、log10函数	23.3节
printf	格式化写 int printf(const char *format, ...); 向stdout流写输出。format指向的字符串说明了后续参数显示的格式。 返回 写入的字符数量。如果发生错误就返回负值。	<stdio.h>
相似函数	fprintf函数、sprintf函数、vfprintf函数、vprintf函数、vsprintf函数	
相关函数	scanf函数	
也可参见	fscanf函数、sscanf函数	3.1节、22.3节
putc	向文件写字符 int putc(int c, FILE *stream); 把字符c写到stream指向的流中。注意: putc函数通常作为宏来实现的。它可能不只计算stream一次。 返回 c(写入的字符)。如果写发生错误, putc函数会设置流的错误指示器,并且返回EOF。	<stdio.h>
相似函数	fputc函数、putchar函数	
相关函数	fgetc函数、getc函数	
也可参见	getchar函数	22.4节
putchar	写字符 int putchar(int c); 把字符c写到stdout流中。注意: putchar函数通常作为宏来实现的。 返回 c(写入的字符)。如果写发生错误, putchar函数设置流的错误指示器,并且返回EOF。	<stdio.h>
相似函数	fputc函数、putc函数	
相关函数	getchar函数	
也可参见	fgetc函数、getc函数	7.3节、22.4节
puts	写字符串 int puts(const char *s); 把s指向的字符串写到strout流中,然后写一个换行符。 返回 如果成功返回非负值。如果写发生错误则返回EOF。	<stdio.h>
相似函数	fputs函数	
相关函数	gets函数	
也可参见	fgets函数	13.3节、22.5节
qsort	排序数组 void qsort(void *base, size_t memb, size_t size, int (*compar)(const void *, const void *)); 对base指向的数组排序。数组有nmemb个元素,每个元素大小为size个字节。compar	<stdlib.h>

621

	是指向“比较函数”的指针。当把指向两个数组元素的指针传递过来时，函数依赖于第一个数组元素是否小于、等于或者大于第二个数组元素，应该返回负数、零或正整数。	
相关函数	bsearch函数	17.7节、26.2节
raise	产生信号	<signal.h>
	int raise(int sig); 产生数为sig的信号。	
返回	如果成功，返回零；否则返回非零值。	
相似函数	abort函数	
相关函数	signal函数	24.3节
rand	产生伪随机数	<stdlib.h>
	int rand(void);	
返回	0到RAND_MAX（包括RAND_MAX在内）之间的伪随机整数。	
相关函数	srand函数	26.2节
realloc	调整内存块	<stdlib.h>
	void *realloc(void *ptr, size_t size); 假设ptr指向先前由calloc函数、malloc函数或realloc函数获得内存块。realloc函数分配size个字节的内存块，并且如果需要还会复制旧内存块的内容。	
返回	指向新内存块开始处的指针。如果无法分配要求尺寸的内存块，那么返回空指针。	
相似函数	calloc函数、malloc函数	
相关函数	free函数	17.3节
remove	移除文件	<stdio.h>
	int remove(const char *filename); 删除文件，此文件名由filename指向。	
返回	如果成功就返回零；否则返回非零值。	
也可参见	rename函数	22.2节
rename	重命名文件	<stdio.h>
	int rename(const char *old, const char *new); 改变文件的名字。old和new指向的字符串分别包含旧的文件名和新的文件名。	
返回	如果改名成功就返回零。如果操作失败，就返回非零值（可能因为旧文件目前是打开的）。	
也可参见	remove函数	22.2节
rewind	返回到文件头	<stdio.h>
	void rewind(FILE *stream); 为stream指向的流设置文件位置指示器到文件的开始处。为流清除错误指示器和文件尾指示器。	
相似函数	fseek函数、fsetpos函数	
相关函数	clearerr函数	
也可参见	feof函数、ferror函数、fgetpos函数、ftell函数	22.7节
scanf	格式化读	<stdio.h>
	int scanf(const char *format, ...); 从stdin流读取任意数量数据项。format指向的字符串说明了读入项的格式。跟随在format后边的参数指向数据项要存储的地方。	
返回	成功读入并且存储的数据项数量。如果发生错误或在可以读入任意数据项之前到达了文件末尾，就返回EOF。	
相似函数	fscanf函数、sscanf函数	
相关函数	printf函数、vprintf函数	
也可参见	fprintf函数、sprintf函数、vfprintf函数、vsprintf函数	3.2节、22.3节

setbuf	设置缓冲区 <stdio.h> void setbuf(FILE *stream, char *buf); 如果buf不是空指针, 那么setbuf的调用就等价于: (void) setvbuf(stream, buf, _IOFBF, BUFSIZ); (BUFSIZ是在<stdio.h>中定义的宏。) 否则, 它等价于: (void) setvbuf(stream, NULL, _IONBF, 0); 相似函数 setvbuf函数 相关函数 fopen函数、freopen函数 也可参见 fflush函数 22.2节
setjmp	准备非局部跳转 <setjmp.h> int setjmp(jmp_buf env); 为了稍候用于longjmp函数调用, 所以把当前外部环境存储到env中。 返回 当直接调用时, 返回为零。当从longjmp函数调用中返回时, 返回非零值。 相关函数 longjmp函数 也可参见 signal函数 24.4节
setlocale	设置地区 <locale.h> char *setlocale(int category, const char *locale); 设置程序的地区部分。category说明哪部分有效。locale指向表示新地区的字符串。 返回 如果locale是空指针, 就返回一个指向与当前地区的category相关的字符串的指针。 否则, 返回一个指向与新地区的category相关的字符串的指针。如果操作失败, 则返回空指针。 相关函数 localeconv函数 25.1节
setvbuf	设置缓冲区 <stdio.h> int setvbuf(FILE *stream, char *buf, int mode, size_t size); 改变由stream指向的流的缓冲。mode的值可以是_IOFBF (满缓冲)、_IOLBF (行缓冲) 或者_IONBF (不缓冲)。如果buf是空指针, 那么若需要则自动分配缓冲区。否则, buf指向用作缓冲区的内存块。size是内存块中字节的数量。 注意: 必须在打开流之后但对流的任何操作执行之前, 调用setvbuf函数。 返回 如果操作成功, 就返回零。如果mode无效或者无法满足要求, 则返回非零值。 相似函数 setbuf函数 相关函数 fopen函数、freopen函数 也可参见 fflush函数 22.2节
signal	安装信号处理函数 <signal.h> void (*signal(int sig, void (*func)(int)))(int); 安装func指向的函数作为数sig的信号处理函数。 返回 指向此信号前一个处理函数的指针。如果无法安装处理函数, 则返回SIG_ERR。 相关函数 abort函数、raise函数 24.3节
sin	正弦 <math.h> double sin(double x); 返回 x的正弦值 (按照弧度衡量的)。 相关函数 acos函数、asin函数、atan函数、atan2函数、cos函数、tan函数 23.3节
sinh	双曲正弦 <math.h> double sinh(double x); 返回 x的双曲正弦值 (按照弧度衡量的)。如果x的数过大, 那么可能会发生取值范围错误。 相关函数 cosh函数、tanh函数 也可参见 acos函数、asin函数、atan函数、atan2函数、cos函数、sin函数、tan函数 23.3节

sprintf	格式串写 <code><stdio.h></code> <code>int sprintf(char *s, const char *format, ...);</code> 与fprintf函数和printf函数很类似，但是sprintf函数不是把字符写入流，而是把字符存储到s指向的数组中。format指向的字符串说明了后续参数显示的格式。在输出的末尾存储一个空字符到数组中。 返回 存储到数组中的字符数量，不计空字符。 相似函数 fprintf函数、printf函数、vfprintf函数、vprintf函数、vsprintf函数 相关函数 sscanf函数 也可参见 fscanf函数、scanf函数 22.8节
sqrt	平方根 <code><math.h></code> <code>double sqrt(double x);</code> 返回 x的平方根。如果x是负数，则会发生定义域错误。 相似函数 pow函数 23.3节
srand	启动伪随机数产生器 <code><stdlib.h></code> <code>void srand(unsigned int seed);</code> 使用seed来初始化由rand函数调用而产生的伪随机序列。 相关函数 rand函数 26.2节
sscanf	格式串读 <code><stdio.h></code> <code>int sscanf(const char *s, const char *format, ...);</code> 与fscanf函数和scanf函数很类似，但是sscanf函数不是从流读取字符，而是从s指向的字符串中读取字符。format指向的字符串说明了读入项的格式。跟随在format后的参数指向数据项要存储的地方。 返回 成功读入并且存储的数据项数量。如果在可以读入任意数据项之前到达了字符串末尾，就返回EOF。 相似函数 fscanf函数、scanf函数 相关函数 sprintf函数、vsprintf函数 也可参见 fprintf函数、printf函数、vfprintf函数、vprintf函数 22.8节
strcat	字符串的连接 <code><string.h></code> <code>char *strcat(char *s1, const char *s2);</code> 把s2指向的字符串连接到s1指向的字符串后边。 返回 s1（指向连接后字符串的指针）。 相似函数 strncat函数 13.5节、23.5节
strchr	搜索字符串中字符 <code><string.h></code> <code>char *strchr(const char *s, int c);</code> 返回 指向字符的指针，此字符是s所指向的字符串的前n个字符中第一个遇到的字符c。如果没有找到c，则返回空指针。 相似函数 memchr函数 也可参见 strpbrk函数、strrchr函数、strstr函数 23.5节
strcmp	比较字符串 <code><string.h></code> <code>int strcmp(const char *s1, const char *s2);</code> 返回 负数、零还是正整数，依赖于s1所指向的字符串是小于、等于还是大于s2所指的字符串。 相似函数 memcmp函数、strcoll函数、strncmp函数 13.5节、23.5节
strcoll	采用指定地区的比较序列进行字符串比较 <code><string.h></code> <code>int strcoll(const char *s1, const char *s2);</code> 返回 负数、零还是正整数，依赖于s1所指向的字符串是小于、等于还是大于s2所指的字符串。根据当前地区的LC_COLLATE类型规则来执行比较操作。

相似函数	memcmp函数、strcmp函数、strncmp函数	
相关函数	strxfrm函数	23.5节
strcpy	字符串复制 char *strcpy(char *s1, const char *s2); 把s2指向的字符串复制到s1所指向的数组中。	<string.h>
返回	s1 (指向目的的指针)。	
相似函数	memcpy函数、memmove函数、strncpy函数	13.5节、23.5节
strcspn	搜索集合中不在初始范围内的字符串 size_t strcspn(const char *s1, const char *s2);	<string.h>
返回	最长的初始字符段的长度, 此初始字符段由s1指向的, 但是不包含s2指向的字符串中的任何字符。	
相关函数	strspn函数	23.5节
strerror	把错误数转换成为字符串 char *strerror(int errnum);	<string.h>
返回	指向字符串的指针, 此字符串含有的出错消息对应errnum的值。	
相关函数	perror函数	24.2节
strftime	把格式化的日期和时间写到字符串中 size_t strftime(char *s, size_t maxsize, const char *format, const struct tm *timeptr);	<time.h>
	在format指向的字符串的控制下把字符存储到s指向的数组中。格式串可能含有不用改变就进行复制的普通字符和转换说明符, 其中转换说明符要用timeptr指向的结构中的值进行替换。maxsize参数限制了可以存储的字符的数量 (包括空字符)。	
返回	如果要存储的字符数量 (包括空字符) 超过了maxsize, 那么返回零; 否则, 返回存储的字符数量 (不包括空字符)。	
相似函数	asctime函数、ctime函数	
相关函数	difftime函数、gmtime函数、localtime函数、mktime函数、time函数	26.3节
strlen	字符串长度 size_t strlen(const char *s);	<string.h>
返回	s指向的字符串长度, 不包括空字符。	13.5节、23.5节
strncat	有限制的字符串的连接 char *strncat(char *s1, const char *s2, size_t n);	<string.h>
	把来自s2所指向的数组的字符连接到s1指向的字符串后边。当遇到空字符或已经复制了n个字符时, 复制操作停止。	
返回	s1 (指向连接后字符串的指针)。	
相似函数	strcat函数	23.5节
strncmp	有限制的字符串比较 int strncmp(const char *s1, const char *s2, size_t n);	<string.h>
返回	负整数、零还是正整数, 依赖于s1所指向的数组的前n个字符是小于、等于还是大于s2所指向的数组的前n个字符。如果在其中某个数组中遇到空字符, 比较都会停止。	
相似函数	memcmp函数、strcmp函数、strcoll函数	23.5节
strncpy	有限制的字符串复制 char *strncpy(char *s1, const char *s2, size_t n);	<string.h>
	把s2指向的数组的前n个字符复制到s1所指向的数组中。如果在s2指向的数组中遇到一个空字符, 那么strncpy函数为s1指向的数组添加空字符直到写完n个字符的总数量。	
返回	s1 (指向目的的指针)。	

相似函数	memcpy函数、memmove函数、strcpy函数	23.5节
strpbrk	为一组字符之一搜索字符串 <code><string.h></code> <code>char *strpbrk(const char *s1, const char *s2);</code> 返回 指向字符的指针，此字符是s1所指向字符串中与s2所指向字符串中的字符相匹配的最左侧的字符。如果没有找到匹配字符，则返回空指针。	
也可参见	memchr函数、strchr函数、strrchr函数、strstr函数	23.5节
strrchr	反向搜索字符串中字符 <code><string.h></code> <code>char *strrchr(const char *s, int c);</code> 返回 指向字符的指针，此字符是s所指向字符串中最后一个遇到的字符c。如果没有找到c，则返回空指针。	
也可参见	memchr函数、strchr函数、strpbrk函数、strstr函数	23.5节
strspn	搜索集合中在初始范围内的字符串 <code><string.h></code> <code>size_t strspn(const char *s1, const char *s2);</code> 返回 最长的初始字符段的长度，此初始字符段是由s1指向的且与s2指向的字符串中的全部字符一致的字符段。	
相关函数	strcspn函数	23.5节
strstr	搜索子字符串 <code><string.h></code> <code>char *strstr(const char *s1, const char *s2);</code> 返回 指针，此指针指向s1字符串中的字符第一次出现在s2字符串中的位置。如果没有发现匹配，就返回空指针。	
也可参见	memchr函数、strchr函数、strpbrk函数、strrchr函数	23.5节
strtod	把字符串转换成双精度数 <code><stdlib.h></code> <code>double strtod(const char *nptr, char **endptr);</code> 函数会跳过nptr所指向的字符串中的空白字符，然后把后续字符都转换为double型的值。如果endptr不是空指针，那么strtod就修改endptr指向的对象，从而使endptr指向第一个剩余字符。如果没有发现double型的值，或者有错误的格式，那么strtod函数把nptr存储到endptr指向的对象中。如果要表示的数过大或者过小，函数就把ERANGE存储到errno中。 返回 转换的数。如果没有转换可以执行，就返回零。如果要表示的数过大，则返回正的或负的HUGE_VAL，这要依赖于数的符号而定。如果要表示的数过小，则返回零。	
相似函数	atof函数	
相关函数	strtol函数、strtoul函数	
也可参见	atoi函数、atol函数	26.2节
strtok	搜索字符串记号 <code><string.h></code> <code>char *strtok(char *s1, const char *s2);</code> 在s1指向的字符串中搜索“记号”。组成此记号的字符不在s2指向的字符串中。如果存在记号，则把跟在记号后边的字符变为空字符。如果s1是空指针，则将继续由strtok函数最近一次调用开始的搜索。在上一个记号尾部的空字符之后立即开始搜索。 返回 指向记号的第一个字符的指针。如果没有发现记号，就返回空指针。	
也可参见	memchr函数、strchr函数、strpbrk函数、strrchr函数、strstr函数	23.5节
strtol	把字符串转换成长整数 <code><stdlib.h></code> <code>long int strtol(const char *nptr, char **endptr, int base);</code> 函数跳过nptr指向字符串中的空白字符，然后把后续字符转换成long int型的值。如果base是2~36之间的数，则把它用作数的基数。如果base为零，除非数是以0（八进制）或者0x/0X（十六进制）开头的，否则就把数设定为十进制的。如果endptr不是空指针，那么strtol函数会修改endptr指向的对象以便endptr可以指向第一个剩余字符。如果没有发现long int型的值，或者它有错误的格式，那么strtol函数会把nptr存储	

	到endptr指向的对象中。如果没有能表示的数，函数会把ERANGE存储到errno中。	
	返回 转换的数。如果没有转换可以执行，则返回零。如果无法表示数，则依赖于数的符号返回LONG_MAX或者LONG_MIN。	
	相似函数 atoi函数、atol函数、strtoul函数	
	相关函数 strtod函数	
	也可参见 atof函数	26.2节
	strtoul 把字符串转换成无符号长整数	<stdlib.h>
	unsigned long int strtoul(const char *nptr, char **endptr, int base);	
	strtuol函数和strtol函数一样，只不过前者会把字符串转换成为无符号长整数。	
	返回 转换的数。如果没有转换可以执行，则返回零。如果无法表示数，则返回ULONG_MAX。	
	相似函数 atoi函数、atol函数、strtol函数	
	相关函数 strtod函数	
	也可参见 atof函数	26.2节
	strxfrm 转换指定地区的字符串	<string.h>
	size_t strxfrm(char *s1, const char *s2, size_t n);	
	函数转换由s2指向的字符串，把结果的前n个字符（包括空字符）放到s1指向的数组中。调用带有两个转换的字符串的strcmp函数应该会产生相同的结果（负数、零或正数），就像调用带有原始字符串的strcoll函数。	
	返回 转换的字符串的长度（可能超过n）。	
631	相似函数 strcmp函数、strcoll函数	23.5节
	System 执行操作系统命令	<stdlib.h>
	int system(const char *string);	
	把string指向的字符串传递给操作系统的命令处理器（命令解释程序）来执行。	
	返回 当string是空指针时，如果命令处理器有效，则返回非零值。如果string不是空指针，则返回由实现定义的值。	
	也可参见 getenv函数	26.2节
	tan 正切	<math.h>
	double tan(double x);	
	返回 x的正切值（按照弧度衡量的）。	
	相关函数 acos函数、asin函数、atan函数、atan2函数、cos函数、sin函数	23.3节
	tanh 双曲正切	<math.h>
	double tanh(double x);	
	返回 x的双曲正切值。	
	相关函数 cosh函数、sinh函数	
	也可参见 acos函数、asin函数、atan函数、atan2函数、cos函数、sin函数、tan函数	23.3节
	time 当前时间	<time.h>
	time_t time(time_t *timer);	
	返回 当前的日历时间。如果日历时间无效，则返回(time_t)-1。如果timer不是空指针，也把返回值存储到timer指向的对象中。	
	相似函数 clock函数	
	相关函数 asctime函数、ctime函数、difftime函数、gmtime函数、localtime函数、mktime函数、strftime函数	26.3节
	tmpfile 创建临时文件	<stdio.h>
	FILE *tmpfile(void);	
	创建临时文件，此文件在被关闭或者程序结束时会被自动删除。按照"wb+"模式打开文件。	
	返回 文件指针。当执行对此文件的后续操作时候用到此指针。如果无法创建文件，则返回空	

	指针。	
相关函数	tmpnam函数、fopen函数	22.2节
tmpnam	产生临时文件名 <code>char *tmpnam(char *s);</code> 产生临时文件名。如果s是空指针，那么tmpnam把文件名存储在静态变量中。否则，它会把文件名复制到s指向的字符数组中。（数组必须足够长可以存储L_tmpnam个字符，这里的L_tmpnam是在<stdio.h>头文件中定义的宏。）	<stdio.h>
返回	指向文件名的指针。	
相关函数	tmpfile函数	22.2节
tolower	转换成小写字母 <code>int tolower(int c);</code> 如果c是大写字母，则返回相应的小写字母。如果c不是大写字母，则返回无变化的c。	<ctype.h>
返回		
相似函数	toupper函数	
相关函数	islower函数、isupper函数	
也可参见	isalpha函数	23.4节
toupper	转换成大写字母 <code>int toupper(int c);</code> 如果c是小写字母，则返回相应的大写字母。如果c不是小写字母，则返回无变化的c。	<ctype.h>
返回		
相似函数	tolower函数	
相关函数	islower函数、isupper函数	
也可参见	isalpha函数	23.4节
ungetc	未读取的字符 <code>int ungetc(int c, FILE *stream);</code> 把字符c回退到stream指向的流中，并且清除流的文件尾指示器。由连续的ungetc函数调用回退的字符数量有变化。只能保证第一次调用成功。调用文件定位函数（fseek函数、fsetpos函数或者rewind函数）会导致回退的字符丢失。	<stdio.h>
返回	c（回退的字符）。如果没有读取操作或者文件定位操作就试图回退过多的字符，那么函数将会返回EOF。	
相关函数	fgetc函数、getc函数、getchar函数	22.4节
va_arg	从可变实际参数列表中获取参数 类型 va_arg(va_list ap, 类型); 从变量参数列表中获取一个参数，然后修改ap使va_arg下一次的使用可以获取后面的参数。在va_arg第一次使用之前必须由va_start对ap进行初始化。	<stdarg.h>
返回	假设参数的类型（在采用了默认的实际参数提升之后）与类型一致，返回参数的值。	
相关函数	va_end函数、va_start函数	
也可参见	vfprintf函数、vprintf函数、vsprintf函数	26.1节
va_end	结束可变实际参数列表的处理 <code>void va_end(va_list ap);</code> 结束与ap相关的可变实际参数列表的处理。	<stdarg.h>
相关函数	va_arg函数、va_start函数	
也可参见	vfprintf函数、vprintf函数、vsprintf函数	26.1节
va_start	开始可变实际参数列表的处理 <code>void va_start(va_list ap, parmN);</code> 必须在访问参数列表之前调用它。初始化ap以便稍后va_arg和va_end的使用。parmN是最后一个普通参数的名字（此参数后边跟着，...）。	<stdarg.h>
相关函数	va_arg函数、va_end函数	

632

633

也可参见		vfprintf函数、vprintf函数、vsprintf函数	26.1节
vfprintf	用可变实际参数列表格式化写文件	<code><stdio.h></code> <pre>int vfprintf(FILE *stream, const char *format, va_list arg);</pre> 函数等价于用arg替换带有可变实际参数列表的fprintf函数。 返回 写入的字符数量。如果发生错误就返回负值。 相似函数 fprintf函数、printf函数、sprintf函数、vprintf函数、vsprintf函数 也可参见 va_arg函数、va_end函数、va_start函数	26.1节
vprintf	用可变实际参数列表格式化写	<code><stdio.h></code> <pre>int vprintf(const char *format, va_list arg);</pre> 函数等价于用arg替换带有可变实际参数列表的printf函数。 返回 写入的字符数量。如果发生错误就返回负值。 相似函数 fprintf函数、printf函数、sprintf函数、vfprintf函数、vsprintf函数 也可参见 va_arg函数、va_end函数、va_start函数	26.1节
vsprintf	用可变实际参数列表格式化写字符串	<code><stdio.h></code> <pre>int vsprintf(char *s, const char *format, va_list arg);</pre> 函数等价于用arg替换带有可变实际参数列表的sprintf函数。 返回 存储的字符数量，但不计空字符。 相似函数 fprintf函数、printf函数、sprintf函数、vfprintf函数、vprintf函数 也可参见 va_arg函数、va_end函数、va_start函数	26.1节
wcstombs	把宽字符串转换成多字节字符串	<code><stdlib.h></code> <pre>size_t wcstombs(char *s, const wchar_t *pwcs, size_t n);</pre> 把宽字符码序列转换成为对应的多字节字符。pwcs指向含有宽字符的数组。多字节字符存储在s指向的数组中。如果遇到存储的空字符或者要存储的多字节字符将超过n个字节的限制，则转换结束。 返回 存储的字节数，不包括空字符。如果遇到一个代码不对应有效多字节字符时，则返回(size_t)-1。 相关函数 mbstowcs函数 也可参见 mblen函数、mbtowc函数、setlocale函数、wctomb函数	25.2节
wctomb	把宽字符转换成多字节字符	<code><stdlib.h></code> <pre>int wctomb(char *s, wchar_t wchar);</pre> 把代码为wchar的宽字符转换成为一个多字节字符。如果s不是空指针，则把结果存储到s指向的数组中。如果s是空指针，则初始化移位状态。 返回 如果s是空指针，则返回非零值或零值，这依赖于多字节字符是否是依赖状态编码的。如果wchar对应一个有效的多字节字符，则返回字符中字节的数量，如果不是这样，则返回-1。 相关函数 mblen函数、mbtowc函数 也可参见 mbstowcs函数、setlocale函数、wcstomb函数	25.2节
<match.h>函数的错误			
定义域错误	参数超出了函数的定义域。如果出现定义域错误，函数的返回值是由实现定义的，并且函数会把EDOM存储到errno中。		
取值范围错误	函数的返回值超出了double型值的取值范围。如果返回值的数太大以致于无法表示（上溢），则函数返回正的或负的HUGE_VAL，这要依赖于正确结果的符号。此外，函数会把ERANGE存储到errno中。如果返回值的数太小以致于无法表示（下溢），则函数返回零。一些实现也可能会把ERANGE存储到errno中。		