

ctype.h 字符处理类

isalnum int isalnum(int c)	字母还是数字
isalpha int isalpha(int c)	是否字母
isctrl int isctrl(int c)	是否控制码
isdigit int isdigit(int c)	是否数字
isgraph int isgraph(int c)	是否为可打印的非空字符
islower int islower(int c)	是否小写字母
isprint int isprint(int c)	是否为可打印字符
ispunct int ispunct(int c)	是否为表示标点符号的字符
isspace int isspace(int c)	是否为空白字符
isupper int isupper(int c)	是否为大写字母
isxdigit int isxdigit(int c)	是否为十六进制数
tolower int tolower(int c)	转换为小写字母
toupper int toupper(int c)	转换为大写字母

math.h 数字类

acos double acos(double arg)	反余弦函数
asin double asin(double arg)	反正弦函数
atan double atan(double arg)	反正切函数
atan2 double atan2(double arg1,double arg2)	带象限的反正切函数
ceil double ceil(double arg)	大于或等于 arg 的最小正整数
cos double cos(double arg)	余弦函数
cosh double cosh(double arg)	双余弦函数
exp double exp(double arg)	指数函数
fabs double fabs(double arg)	双精度的浮点绝对值
floor double floor(double arg)	大于或等于 arg 的最大正整数
fmod double fmod(double arg1,double arg2)	浮点数的余数
frexp double frexp(double arg1,int *arg2)	将浮点数分为两部分
ldexp(double arg1,int arg2)	乘以 2 的幂
log double log(double arg)	自然对数函数
log10 double log10(double arg)	以 10 为底的对数函数
modf double modf(double value,double *iptr)	拆开为整数部分和小数部分
pow double pow(double arg1,double arg2)	求幂函数
sin double sin(double arg)	正弦函数
sinh double sinh(double arg)	双曲正弦
sqrt double sqrt(double arg)	平方根函数
tan double tan(double arg)	正切函数
tanh double tanh(double arg)	双曲正切函数

setjmp.h 非局部跳转

longjmp void longjmp(jmp_buf env,int val)	长跳转
---	-----

setjmp int setjmp(jmp_buf env)	设置返回点跳转
stdio.h 输入和输出类函数	
getchar int getchar(void)	获得字符
gets char *gets(char *s)	读字符串
printf int printf(const char *format, ...)	写格式化数据
putchar int putchar(int value)	写字符函数
puts int puts(const char *s)	写字符串函数
scanf int scanf(const char *format, ...)	读格式化数据
sprintf int sprintf(char *s, const char *format, ...)	将格式化数据写入字符串
sscanf int sscanf(const char *s, const char *format, ...)	从字符串中读取格式化数据
stdlib.h 通用子程序类	
abort void abort(void)	非正常结束程序
abs int abs(int j)	绝对值函数
atof double atof(const char *nptr)	转换 ASCII 为双精度
atoi int atoi(const char *nptr)	转换 ASCII 为整数
atol long atol(const char *nptr)	转换 ASCII 为长整形
bsearch void *bsearch(const void *key, const void *base, size_t nmemb, size_t size, int (*compare)(const void *_key, const void *_base));	在数组中搜索
calloc void *calloc(size_t nelem, size_t elsize)	为目标数组分配存储器单元
div div_t div(int numer, int denom)	除法运算函数
exit void exit(int status)	结束程序
free void free(void *ptr)	释放存储器单元
labs long int labs(long int j)	整形数取绝对值
ldiv ldiv_t ldiv(long int numer, long int denom)	长整形除法
malloc void *malloc(size_t size)	分配存储器
qsort void qsort(const void *base, size_t nmemb, size_t size, int (*compare)(const void *_key, const void *_base));	数组排序
rand int rand(void)	随机数生成函数
realloc void *realloc(void *ptr, size_t, size)	重新分配存储器单元函数
srand void srand(unsigned int seed)	设置随机数 (的种子)
strtod double strtod(const char *nptr, char **endptr)	将字符串转换为双精度数
strtol long int strtol(const char *nptr, char **endptr, intase)	将字符串转换为长整形数

strtoul unsigned long int strtoul 字符串转换为无符号整形数
 const char *nptr, char
 **endptr, base int)

string.h 字符串处理类

memchr void *memchr(const void *s, int c, size_t n) 在存储器中搜索字符
memcmp int memcmp(const void *s1, const void *s2, size_t n) 比较存储器内容
memcpy void *memcpy(void *s1, const void *s2, size_t n) 拷贝存储器内容
memmove void *memmove(void *s1, const void *s2, size_t n) 移动存储器内容
memset void *memset(void *s, int c, size_t n) 置存储器
strcat char *strcat(char *s1, const char *s2) 逻辑字符串
strchr char *strchr(const char *s, int c) 在字符串中找某一个字符
strcmp int strcmp(const char *s1, const char *s2) 比较两个字符串
strcoll int strcoll(const char *s1, const char *s2) 比较字符串
strcpy char *strcpy(char *s1, const char *s2) 拷贝字符串
strcspn size_t strcspn(const char *s1, const char *s2) 在字符串中跨过被排除的字符
strerror char *strerror(int errnum) 给出一个错误信息字符串
strlen size_t strlen(const char *s) 计算字符串长度函数
strncat char *strncat(char *s1, const char *s2, size_t n) 将指定数量的字符与字符串连接起来
strncmp int strncmp(const char *s1, const char *s2, size_t n) 将指定数量的字符与字符串相比较
strncpy char *strncpy(char *s1, const char *s2, size_t n) 在字符串中复制指定的字符
strpbrk char *strpbrk(const char *s1, const char *s2) 在字符串中寻找任何指定的字符
strrchr char *strrchr(const char *s, int c) 从字符串的右端开始寻找字符
strspn size_t strspn(const char *s1, const char *s2) 在字符串中统计和分析字符
strstr char *strstr(const char *s1, const char *s2) 在字符串中搜索子字符串
strtok char *strtok(char *s1, const char *s2) 将标志前的字符剪掉
strxfrm size_t strxfrm(char *s1, const char *s2, size_t n) 转换字符串并返回其长度