C++常用函数

PDF

有些函数如果碰到了再加进来。

    - exp
    - fabs
    - floor
    - fmod
    - hypot
    - log
    - log10
    - max
    - min
    - modf
    - pow
    - rand
    - sin，sinh
    - sqrt
    - srand
    - tan，tanh
- 输入和输出函数
    - freopen
    - getchar
    - gets
    - putchar
    - puts
- 进程控制函数
    - exit
- 字符串操作函数
    - strcat
    - strchr
    - strcmp
    - strcpy
    - stricmp
    - strlen
    - strlwr
    - strncmp

- strncpy
- strrev
- strstr

缓冲区操作函数

- memchr
- memcpy
- memicmp
- memmove
- memset
- swab

---

memchr

函数原型： void *memchr(const void *buf，int c，sizet count)；
参数： buf 缓冲区的指针；c 查找的字符；count 检查的字符个数。
所需头文件： cstring
功能： 查找 buf 的前 count 个字节中 c 的第一次出现，当找到 c 或已检查完 count 个字节时停止。
返回值： 如果成功，返回 buf 中 c 首次出现的位置的指针；否则返回 NULL

memcpy

函数原型： void *memcpy(void *dest，const void *src，sizet count)；
参数： dest 目的缓冲区；src 源缓冲区；count 拷贝的字符个数。
所需头文件： cstring
功能： 从 src 拷贝 count个字节到dest。如果源缓冲区和目的缓冲区重叠，这个函数不能保证正确拷贝；对于这种情况可使用 memmove 处理。
返回值： 返回 dest 的值。

memicmp

函数原型： int memicmp(const void *buf1，const void *buf2，unsigned intcount)；
参数： bufl 第一个缓冲区；buf2第二个缓冲区；count 字符个数。
所需头文件： cstring
功能： 比较两个缓冲区 bufl 和 buf2的前 count 个字符，比较过程是大小写无关的。
返回值： bufl 和 buf2的前 count 个字节之间的关系：
<0：bufl 小于 buf2； =0：bufl 等于 buf2；>0：bufl 大于 bur2

memmove

函数原型： void *memmove(void *dest，const void *src，sizet count)；
参数： dest 目的对象；src 源对象；count 拷贝的字符字节个数。
所需头文件： cstring
功能： 从 src 拷贝 count 个字节到 dest。如果源区域与目的区域有重叠，memmove 也能确保正确拷贝。
返回值： 返回 dest 的值。

memset

函数原型： void *memset(void *dest，int c，sizet count)；
参数： dest 目的指针；c 设置的字符；count 字符个数。
所需头文件： cstring
功能： 设置 dest 的前 count 个字节为字符 c。
返回值： 返回 dest 的值。

swab

函数原型： void swab(char *src，char *dest，int n)；
参数： src 需拷贝和交换的数据；dest 交换结果数据的存储位置；n 拷贝和交换的字节个数。

所需头文件： cstdlib
功能： 从 src 拷贝 n 个字节，交换每对相邻的字节，并把结果存储在dest中。一般用于为转换到使用不同字节次序的机器上而准备二进制数据。
返回值： 无

字符分类函数

- isalpha
- islower
- isupper

---

isalpha

函数原型： int isalpha(int c)；
所需头文件： cctype
功能： 测试 c 是否字母。
返回值： 如果 c 在 A～Z 或 a～z 的范围内，则返回一个非0值；否则返回0。

islower

函数原型： int islower(int c)；
所需头文件： cctype
功能： 测试是否小写字母。
返回值： 如果 c 是一个小写字母(a～z)

isupper

函数原型： int isupper(int c)；
所需头文件： cctype

功能： 测试是否大写字母。
返回值： 如果 c 是一个大写字母，则返回一个非0值；否则返回0。

数据转换函数

---

abs

函数原型： int abs(int n)；
参数： n 整数值。
所需头文件： cstdlib
功能： 求绝对值。
返回值： 返回 n 的绝对值。

atof，atoi，atol

函数原型：
double atof(const char *string)；
int atoi(const char *string)；
long atol(const char *xstring)；
参数： string 要转换的字符串。
所需头文件： cstdlib
功能： 将字符串转换成 double(atof)、integer(atoi)或 long(ato1)型数据。
返回值： 返回转换后的结果值，如果输入不能转换成对应类型的值，返回值为0.0(atof)或0(atoi，atol)。溢出情况下返回值不确定。

ecvt

函数原型： char *ecvt (double value，int count，int dec，int *sign)；
参数： value 被转换的数；count 存储的数字个数；dec 存储的小数点位置；sign 转换的数的符号。
所需头文件： cstdlib
功能： 将 double 型浮点数转换成指定长度的字符串，返回值：返回数字字符串的一个指针；没有错误返回

tolower

函数原型： int tolower(int c)；
参数： c 要转换的字符。
所需头文件： cstdlib和cctype
功能： 将字符转换为小写字母。
返回值： 返回转换结果。

toupper

函数原型： int toupper(int c)；
参数： c 要转换的字符。
所需头文件： cstdlib和cctype
功能： 将字符转换为大写字母。
返回值： 返回转换结果。

数学函数

- abs
- acos
- asin

- atan，atan2
- atof
- ceil
- cos，cosh
- difftime
- exp
- fabs
- floor
- fmod
- hypot
- log
- log10
- max
- min
- modf
- pow
- rand
- sin，sinh
- sqrt
- srand
- tan，tanh

---

abs

函数原型： int abs(int n);
参数： in 需要求绝对值的整数。
所需头文件： cstdlib或cmath
功能和返回值： 返回 n 的绝对值；没有错误返回

acos

函数原型： double acos(double x) ;
参数： x 是-1到 l 之间的值。
所需头文件： cmath
功能和返回值： 计算并返回范围在0到 π 弧度之间的 x 的反余弦值。

asin

函数原型： double asin(double x)；
参数： x 是-l 到1之间的值。
所需头文件： cmath
功能和返回值： 计算并返回范围在-π/2到 π/2弧度之间的 x 的反正弦值。

atan，atan2

函数原型：
double atan(double x)；
double atan2(double y，double x)；
所需头文件： cmath
功能： 计算 x(atan)或 y／x(atan2)的反正切值。
返回值： atan 返回 x 的反正切值，atan2返回 y／x 的反正切值。如果 x 为0，则 atan 返回0。如果 atan2 的两个参数都为0，该函数返回0。

atof

函数原型： double atof(const char ustring)；
参数： string 需要转换的字符串。
所需头文件： cmath或cstdlib
功能和返回值： 将字符串转换成 double 值并返回该值。如果 string 不能转换成 double 类型的值，返回值为0.0。

ceil

函数原型： double ceil(double x)；
所需头文件： cmath
功能： 对 x 向上取整，并以 double 型浮点数形式存储结果。
返回值： 返回一个 double 型的大于或等于 x 的最小整数；没有错误返回。

cos，cosh

函数原型： double cos(double x)；
参数： x 弧度值。
所需头文件： cmath
功能和返回值： 计算并返回 x 的余弦值(cos)或双曲余弦值(cosh)。

difftime

函数原型： double difftime(time_t time2, time_t time1)；
参数： time2 终止时间；time1 开始时间。
所需头文件： ctime
功能： 计算两个指定时间值之间的差。
返回值： 返回从 time1 到 time2 之间经过的时间

exp

函数原型： double exp(double x)；
所需头文件： cmath
功能和返回值： 计算并返回 e 的 x 次幂。

fabs

函数原型： double fabs(double x)；
所需头文件： cmath
功能和返回值： 计算并返回浮点参数 x 的绝对值。

floor

函数原型： double floor(double x)；
所需头文件： cfloat
功能： 向下取整，并以 double 型浮点数形式存储结果。
返回值： 返回一个 double 型的小于或等于 x 的最大整数；没有错误返回。

fmod

函数原型： double fmod(double x，double y)；
所需头文件： cmath
功能和返回值： 计算并返回 x／y 的余数，如果 y 值是0．0，返回一个静止 NaN。

hypot

函数原型： double hypot (double x，double y)；
参数： 直角三角形的两个直角边长度。
所需头文件： cmath
功能和返回值： 计算并返回直角三角形的斜边长度(x 与 y 的平方根)，上溢出时返 INF(无穷大)。

log

函数原型： double log(double x)；
所需头文件： cmath
功能和返回值： 计算并返回 x 的自然对数。如果 x 是负数，返回值不确定。如果 x 为0，返回 INF(无穷大)。

log10

函数原型： double log10(double x)；

所需头文件： cmath
功能和返回值： 汁算并返回 x 的以10为底的对数。如果 x 是负数，返回值不确定。如果 X 为0，返回 INF(无穷大)。

max

函数原型： type max (type a，type b)；
参数： type 任何数值数据类型；a 和 b 是参与比较的两个数，必须是相同类型。
所需头文件： cstdlib
功能和返回值： 比较 a 和 b 并返回其中较大者。

min

函数原型： type min (type a，type b)；
参数： type 任何数值数据类型。a 和 b 是参与比较的两个数，必须是相同类型。
所需头文件： cstdlib
功能和返回值： 比较 a 和 b 并返回其中较小者。

modf

函数原型： double modf(double x，double *inptr)；
参数： x 需要分解的数；lnptr 指向分解后整数部分的指针。
所需头文件： cmath
功能和返回值： 将浮点值 x 分解成小数和整数部分，每个都与 x 具有同样的符号。返回 x 的带符号的小数部分，整数部分作为浮点值存储在 intptr 处。

pow

函数原型： double pow(double x，double y)；
所需头文件： cmath
功能和返回值： 计算并返回 x 的 y 次幂。

rand

函数原型： int rand(void)；
所需头文件： cstdlib
功能和返回值： 返回一个 0 ~ RAND_MAX 的随机数

sin，sinh

函数原型：
double sin(double x)；
double sinh(double x)；
参数： x 弧度值。
所需头文件： cmath
功能和返回值： sin 返回 x 的正弦值。sinh 返回 x 的双曲正弦值。

sqrt

函数原型： double sqrt(double x)；
所需头文件： cmath
功能和返回值： 计算并返回 x 的平方根。

srand

函数原型： void srand(unsigned int seed)；
参数： seed 产生随机数的种子。
所需头文件： cstdlib
功能： 为使 rand()产生一序列伪随机整数而设置起始点。使用1作为 seed 参数，可以重新初始化 rand()。

tan，tanh

函数原型：
double tan(double x)；
double tanh(double x)；
参数： x 弧度值。
所需头文件： cmath
功能和返回值： tan 返回 x 的正切值。tanh 返回 x 的双曲正切值。

输入和输出函数

- freopen
- getchar
- gets
- putchar
- puts

---

freopen

函数原型： FILE *freopen(const char *path，const char *mode，FILE *stream)；
参数： path 新文件的路径；mode 文件访问许可；stream FILE 结构的指针。
所需头文件： cstdio
功能： 关闭当前与 stream 关联的文件，并将 stream 重新赋给由 path 指定的文件。
返回值： 返回最新打开的文件的指针。如果出现错误，最初的文件被关闭并返回 NULL 指针值。

getchar

函数原型： int getchar(void)；
所需头文件： stdio．h
功能和返回值： 从 stdin 读取一个字符并返回所读字符，当出现读错误或遇到文件结尾时返回 EOF。

gets

函数原型： char *gets(char *buffer)；
参数： buffer 输入字符串的存储位置。
所需头文件： cstdio
功能： 从标准输入流 stdin 读取一行，并存储在 buffer 中。该行由直到第一个换行符(‘\n’)的所有字符组成，并包括该第一个换行符，然后 gets 在返回该行之前用空字符(‘\0’)代替换行符。
返回值： 如果成功，返回 buffers 如果有错误或遇到文件结尾则返回 NULL 指针。

putchar

函数原型： int putchar(int c)；
参数： c 要写的字符。
所需头文件： cstdio
功能： 写一个字符到 stdout 中。
返回值： 返回所写的字符；如果出现错误，返回 EOF。

puts

函数原型： int puts(const char *string)；
参数： string 要输出的字符串。
所需头文件： cstdio
功能： 将 string 写到标准输出流 stdout，在输出流中用换行符（‘\n’）代替字符串的结尾的空字符（‘\0’）。
返回值： 如果成功，返回一个非负值；如果失败，返回 EOF。

进程控制函数

- exit

exit

函数原型： void exit(int status)；
参数： status 退出状态。
所需头文件： cstdlib
功能： 终止进程。

字符串操作函数

- strcat
- strchr
- strcmp
- strcpy
- stricmp
- strlen
- strlwr
- strncmp
- strncpy
- strrev
- strstr

---

strcat

函数原型： char *strcat(char *strDestination，const char *strSource)；
参数： strDestination 以空字符结尾的目的字符串 strSource 以空字符结尾的源字符串。
所需头文件： cstring
功能： 将 strSource 添加到 strDestination，并用一个空字符结束该结果字符串。用 strSource 的首字符覆盖 strDestination 的结尾空字符。当字符串被拷贝或添加时不执行上溢出检测。如果源和目的字符串重叠，strcat 的行为是不确定的。

返回值： 返回目的字符串。

strchr

函数原型： char *strchr(const char *string，int c)；
参数： string 以空字符结尾的源字符串口要查找的字符。
所需头文件： cstring
功能： 查找 string 中 c 的第一次出现，在查找中包括结尾的空字符。
返回值： 返回 string 中第一次出现的指针；如果 c 未找到，则返回 NULl。

strcmp

函数原型： int strcmp(const char *string1，const char *string2)
参数： stringl，string2被比较的以空字符结尾的字符串。
所需头文件： cstring
功能： 按词典顺序比较 stringl 和 string2，并返回一个值指出它们之间的关系。
返回值： 返回值<0，stringl 小于 string2；返回值 = 0，stringl 等于 string2；返回值〉0，stringl 大于 string2。

strcpy

函数原型： char*strcpy(char *strDestination，const char *strSource)
参数： strDestination 目的字符串;strSource 以空字符结尾的源字符串。
所需头文件： cstring
功能： 把源字符串 strSource(包括结尾的空字符) 拷贝到 strDestination 所指的位置。在字符串被拷贝或添加时不执行上溢出检测。如果源和目的字符串重叠，strcpy 的行为是不确定的。
返回值： 返回目的字符串，没有用于指出错误的返回值

stricmp

函数原型： int stricmp(const char *string1，const char *string2);

参数： string1，string2要比较的以空字符结尾的字符串。
所需头文件： cstring
功能： 忽略大小写来比较两个字符串。_strlcmp 函数以词典次序比较 stringl 和 string2 的小写版本，并返回一个值指出它们之间的关系。
返回值： 返回值<0，stringl 小于 string2；返回值 = 0，stringl 等于 string2；返回值>0，stringl 大于 string2。

strlen

函数原型： sizet strlen(const char*string)；
参数： string 以空字符结尾的字符串。
所需头文件： cstring
功能和返回值： 返回 string 中的字符个数，不包括尾部 NULL。没有指出错误的返回值。

strlwr

函数原型： char* strlwr(char *string)；
参数： string 需要转换成小写的以空字符结尾的字符串。
所需头文件： cstring
功能： 将 string 中的任何大写字母转换成小写，其它字符不受影响。
返回值： 返回转换后的字符串的指针。因为不修改位置的指针相同。没有返回值指出错误。

strncmp

函数原型： int strncmp(const char *stringl，constchar *string2，size_t count)
参数： stringl，string2比较的字符串；count 比较的字符的个数。
所需头文件： cstring
功能： 按词典顺序比较 stringl 和 string2的前 count 个字符，并返回一个值指出串之间的关系。大小写敏感。
返回值： <0，stringl 串小于 string2串。=0，stringl 串等于 string2串；>0，stringl 大于 string2串。

strncpy

函数原型： char *strncpy(char *strDest，const char *strSource，size_t count)
参数： strDest 目的字符串；strSource 源字符串；count 拷贝的字符个数。
所需头文件： cstring
功能： 将 strSource 的前 count 个字符拷贝到 strDest 中并返回 strDest。如果 count 小于或等于 strSource 的长度，空字符不自动添加到拷贝的字符串中。如果 count 大于 strSource 的长度，目的字符串用空字符填充直到 count 的长度。如果源和目的字符串重叠，则 strncpy 的行为是不确定的。
返回值： 返回 strDest。没有返回值则表明出错。

strrev

函数原型： char *strrev(char *string)；
参数： string 要逆转的以空字符结尾的字符串。
所需头文件： cstring
功能： 将 string 中字符反序排列。结尾的空字符保留在原位置。
返回值： 返回改变后的字符串的指针。没有返回值则说明出错。

strstr

函数原型： char*strstr(constchar*string，constchar*strCharSet)；
参数： string 要在其中进行查找的以空字符结尾的字符串;strCharSet 要查找的以空字符结尾的字符串。
所需头文件： cstring
功能和返回值： 返回 strCharSet 在 string 中第一次出现的起始地址，如果 strCharSet 不在 string 中出现，则返回 NULL。