（C 语言）常见字符函数和字符串函数

头文件:<cstring> 或者<string.h> //两种写法选择一个即可,前者是 C++的写法,后者是 C 语言的表示方法

假设定义 char str[1005] , c ;

输入方法:(一般只采用 1、2、5 三种方式输入即可)

//以下 c 语言的输入输出头文件为:<cstdio>或<stdio.h>,

//C++的输入输出方式的头文件在<iosteam>

1、cin>>str;

C++的标准输入,只能输入一个单词或者字母 ， 遇到空格或者回车键结束.

2、scanf(“%s”,str);

C 语言的输入方式,只能输入一个单词或者字母 ， 遇到空格或者回车键结束.

3、gets(str);

可以输入一整行,遇到回车键结束.(最新 C 标准已经删除该方法,故不建议使用这种方式)

4、c=getchar();

仅输入单个字符;

5、cin.getline(str,1000, ‘#’)

cin.getline(str,1000)

说明:

输入一整行.
第一个参数表示要存储到哪个地方;
第二个参数是表示最多接收几个字符;
第三个参数是表示输入的结束符号(当遇到给符号时候输入才结束),该参数如果省略不写,则默认是以回车键结束,如果这个参数不是回车换行键是其他字符时候,那么可以输入多行(即遇到回车键不会输入结束,直到遇到规定字符输入才结束;
第二和第三个参数哪个条件先达到即输入结束.

输出方法:

1、Cout

C++的标准输出,从开始地址直到遇到字符串结束符终止.

2、printf(“%s”,str)

C 的标准输出,从开始地址直到遇到字符串结束符终止.

3、Puts(str)

输出整个字符串+换行;

4、Putchar(c)

输出单个字符;

strlen

size_t strlen(const char* str)

功能：计算字符串长度，不包含’\0’

返回值：返回字符串的字符数

说明：

strlen() 函数计算的是字符串的实际长度，遇到第一个’\0’结束；
参数指向的字符串必须以 ’ \0 ‘结束
函数返回值一定是 size_t ，是无符号的
如果你只定义没有给它赋初值，这个结果是不定的，它会从首地址一直找下去，直到遇到’\0’停止
sizeof 返回的是变量声明后所占的内存数，不是实际长度，此外 sizeof 不是函数，仅仅是一个操作符，strlen()是函数

strcpy

char* strcpy(char* dest,char* src)

功 能: 将参数 src 字符串拷贝至参数 dest 所指的地址

返回值: 返回参数 dest 的字符串起始地址

说明:

源字符串必须以’\0’结束
会将源字符串的’\0’拷贝到目标空间
目标空间必须可变
如果参数 dest 所指的内存空间不够大，可能会造成缓冲溢出的错误情况，在编写程序时需特别留意，或者用 strncpy()来取代

strncpy

char* strncpy(char* dest,const char* src,size_t num)

功能：拷贝 src 字符串的前 num 个字符至 dest

返回值：dest 字符串起始地址

说明:

如果 src 字符串长度小于 num，则拷贝完字符串后，在目标后追加 0，直到 num 个
strncpy 不会向 dest 追加’\0’
src 和 dest 所指的内存区域不能重叠，且 dest 必须有足够的空间放置 n 个字符

strcat

char* strcat(char* dest,const char* src)

功能： 字符串拼接

返回值：返回 dest 字符串起始地址

说明:

- 源字符串必须’\0’结束
- 目标空间必须可修改
- strcat() 会将参数 src 字符串复制到参数 dest 所指的字符串尾部
- dest 最后的结束字符’\0’会被覆盖掉，并在连接后的字符串的尾部再增加一个’\0’
- dest 与 src 所指的内存空间不能重叠，且 dest 要有足够的空间来容纳要复制的字符串

strncat

char* strncat (char* dest,const char* src,size_t num)

功能：将 n 个字符追加到字符串结尾

返回值：返回 dest 字符串的起始地址

说明：

strncat 将会从字符串 src 的开头拷贝 n 个字符到 dest 字符串尾部
dest 要有足够的空间来容纳要拷贝的字符串
如果 n 大于字符串 src 的长度，那么仅将 src 全部追加到 dest 的尾部

strncat 会将 dest 字符串最后的’\0’覆盖掉，字符追加完成后，再追加’\0’

strcmp

int strcmp (const char* str1,const char* str2)

功能：字符串比较

返回值：若参数 s1 和 s2 字符串相同则返回 0，s1 若大于 s2 则返回大于 0 的值，s1 若小于 s2 则返回小于 0 的值

说明：

判断两个字符串大小 1）ASII 码 2）长度
区分大小写比较的，如果希望不区分大小写进行字符串比较，可以使用 stricmp 函数

strncmp

int strncmp(const char* str1,const char* str2,size_t num)

功能：指定长度比较

返回值：同 strcmp

strstr

char* strstr(const char* str,const char* substr)

功能：检索子串在字符串中首次出现的位置

返回值：返回字符串 str 中第一次出现子串 substr 的地址；如果没有检索到子串，则返回 NULL

strchr

char* strchr(const char* str,char c)

功能：查找字符串中第一次出现 c 的位置

返回值：返回首次出现 c 的位置的指针，如果 s 中不存在 c 则返回 NULL

strrchr

char* strchr(const char* str,char c)

功能：查找字符串中最后一次出现 c 的位置

返回值：返回最后一次出现 c 的位置的指针，如果 s 中不存在 c 则返回 NULL

strspn

size_t strspn(const char* str1, const char* str2)

功能：用来计算 str1 字符串开始部分匹配 str2 字符串的字符个数

返回值：返回字符串 str1 开头连续包含字符串 str2 内的字符数目。

如果 str 所包含的字符都属于 str2，那么返回 str1 的长度
如果 str 的第一个字符不属于 str2，那么返回 0

strcspn

size_t strspn(const char* str1, const char* str2)

功能：用来计算 str1 字符串开始部分不匹配 str2 字符串的字符个数

返回值：返回字符串 str1 开头部分不出现在字符串 str2 内的字符数目。

strtok

char* strtok(char* str,const char* sep)

功能：根据分隔符将字符串分隔成一个个片段

返回值：返回下一个分割后的字符串指针，如果已无从分割则返回 NULL

说明：

- sep 参数是个字符串，定义了用作分隔符的字符集合
- 第一个参数指定一个字符串,它包含了一个或者多个由 sqp 字符串中一个或者多个字符分割的标记
- 第一次调用时将字符串首地址传进去，之后调用不用传地址，内部会有 static 函数保存指向地址
- 分隔符不作为输出内容，只做分界符
- 当 strtok 在参数 s 的字符串中发现到参数 sep 的分割字符时则会将该字符改为’\0’字符
- 在第一次调用时，strtok 必须赋予参数 str 字符串，往后的调用则将参数 s 设置成 NULL
- strtok 会修改原字符串，所以必须放至栈上

strerror

char* strerror(int errnum)

功能：返回指向错误信息字符串的指针

说明：

必须包含头文件 errno.h

strpbrk

char* strpbrk(const char* str1, const char* str2)

功能：在字符串 s1 中寻找字符串 s2 中任何一个字符相匹配的第一个字符的位置，空字符 NULL 不包括在内

atoi

int atoi(const char* str)

功能：将字符串转换成整型

返回值：返回转换后的整型数。如果 str 不能转换成 int 或者 str 为空字符串，那么将返回 0

说明：

ANSI C 规范定义了 stof()、atoi()、atol()、strtod()、strtol()、strtoul()共 6 个可以将字符串转换为数字的函数，可以对比学习
另外在 C99/C++11 规范中又新增了 5 个函数，分别是 atoll()、strtof()、strtold()、strtoll()、strtoull()

字符分类函数：

int isalnum(int c)：检查字符是否为数字或字母；(0~9,a~z,A~Z)

int isalpha(int c)：检查字符是否为字母；(a~z, A~Z)

int iscntrl(int c)：检查字符是否为控制字符；(八进制 000~037 以及 177 的字符)

int isdigit(int c)：检查字符是否为十进制数字；(0~9)

int isgraph(int c)：检查字符是否为图形表示，依赖于使用语言的环境；0~9,a~z,A~Z,以及标点符号)

int islower(int c)：检查字符是否为小写的字母；(a~z)

int isprint(int c)：检查字符是否为可打印的；(数字、字母、标点符号、空白字符)

int ispunct(int c)：检查字符是否为标点符号；(! " # $ % & ' ( ) * + , - . / : ; < = > ? @ [ ] ^ _ ` { | } ~等)

int isspace(int c)：检查字符是否为空白字符；(TAB、换行、垂直 TAB、换页、回车、空格)

int isupper(int c)：检查字符是否为大写字母；(A~Z)

int isxdigit(int c)：检查字符是否为十六进制数字；(0 1 2 3 4 5 6 7 8 9 A B C D E F a b c d e f)

字符转换

int tolower(int c)：转化字符为小写字母；

int toupper(int c)：转化字符为大写字母；