**关于 cin**

cin 的用法与 scanf( ) 的用法类似；

1.如果缓冲区里只剩下换行符，则 cin 将跳过缓冲区；

2.如果缓冲区里还有字符 则 cin 将继续读入；

3.当只剩下换行符时 就算是 char 型也要重新输入(不能输入换行符）；

4.输入完毕时 读取后将把换行符 或者把多余的字符（char 型），留在（流）中；

5.当cin读取出错时，将cin 输入为一个垃圾值，并设置失效位；

6.cin 不能读取 换行符 空字符 ，遇到时将跳过；所以不能用 cin 接受换行符跟空字符；可以用 cin.get ();

**关于cin.get()**

1． cin.get ( str , 10 ): 接受一个字符串 ； 并将结尾的换行符保留在缓冲区 （流）中；

2．cin.get ( ch ); 接受一个字符从缓冲区的首位置开始包括 换行符

3.当输入字符超过了指定字符 n-1 时，将把前 n-1 个字符输入到制定空间，剩余字符保留到流中；

4.cin.get()只能调用 char 类型,不能调用 int 即其他；

5.缓冲区剩余的内容将不受 cin.get() 限制，而是通过后面的输入函数进行与之匹配的转换

6.当读入一个空行时 将设置失效位；

**关于cin.getline ( )**

cin.get( )与cin.getline( ) 基本相同 但也有不少区别

1.cin.getline ( ) 读取完数据后，不在缓存区留下换行符；

2.当输入字符超过了指定字符 n-1 时, cin.getline ( )将把前 n-1 个字符输入到制定空间，

剩余字符保留到流中

**读取空行和其他问题**

1.当getline()或get()读取空行时将如何？一般是下一条输入语句将在前一条结束读取的位置开始读取, 但当有空行时，当get()读取空行后将设置失效位(failbit)，这意味接下来的输入将被阻断，但可 以用下面的命令来恢复输入：cin.clear();

2.另一个潜在的问题是，输入字符可能比分配的空间长。如果输入行包含的字符数比指定的多，则这get()和getline()将把余下的字符留在输入队列中，而getline()还会设置失效位，并关闭输入。

3. int age = 0;

char address[size] = {0};

cout<<"Enter your age:";

cin>>age;

cout<<"Enter your address:";

cin.getline(address,size);

cout<<"your age is "<<age <<", your address is "<<address<<endl;

上面的写法有问题，会导致第二个输入没有机会。原因是：当cin读取年龄时，将回车键生成的换行符留存了输入队列中，后面的cin.getline()看到换行符后，将认为是一个空行，并将一个空字符串赋给address 数组。解决办法为在cin>>age;后加cin.get();