If ( (year%4==0&&year%100!=0) || year%400==0)

判断是否为闰年

\t 为长度为8的制表符

在没有输入时，getchar返回一个特殊字符EOF

有缺省参数的形参必须连续存在于形参列表的最右。

静态数据成员：类的每个成员共享。（例如点类中点的个数）初始化可在类外完成，用类名限定：例如 int A：：count=10；原类里：static int count；

静态函数成员：可以直接通过类名来调用。

类的友元：处理某几个类成员关系时使用，（例如点类两点的距离），不同类或对象的成员函数之间、类的成员函数与一般函数之间进行数据共享的机制。

友元函数（friend）：定义在类外，定义时无需写类名；可以直接访问该类的私有和保护成员。

友元类：a为b友元类，则a所有成员函数都是b的友元函数。

常成员函数：void abc（）const；常对象：const Point a；常对象只能调用常函数

常函数不能修改目标成员，一般用于输出之类的地方

常成员函数不能调用非常成员函数；

常引用：const Point &p1；可以引用常对象；

派生类：

Class derived：public base1，private base2

｛

Public：

Derived（）

~derived（）

｝

公有继承：可直接访问基类的私有和保护成员。

私有继承：基类的工友和保护成员都以私有的形式出现在派生类中

保护继承：公有和保护成员都以保护成员身份出现在派生类。

虚基类：派生类的几个基类来自于同一个基类，就声明基类为派生类的虚基类。

写法class 派生类名：virtual 继承方式 基类名。

虚函数：当基类和派生类有重名函数时，通过在主函数写一个基类的指针，将其指向一个派生类的元素，就可以通过这个指针访问派生类所继承的基类里面的重名函数，而避免了覆盖。

C 库函数 void \*calloc(size\_t nitems, size\_t size) 分配所需的内存空间，并返回一个指向它的指针。malloc 和 calloc 之间的不同点是，malloc 不会设置内存为零，而 calloc 会设置分配的内存为零。

C 库函数 void \*realloc(void \*ptr, size\_t size) 尝试重新调整之前调用 malloc 或 calloc 所分配的 ptr 所指向的内存块的大小。

New的用法：

const int M = 10, N = 5;//10行5列。  
int \*\* a;  
a = new int \*[M];  
for(int i = 0; i < M; i ++)  
a[i] = new int[N];