1. operator<<重载函数为什么不能定义为成员函数？请整理并阐述，拒绝单纯的复制粘贴。

一个类重载了<<,是为了用<<直接输出类中的对象，例如cout输出这个函数，并不在C++默认函数类中，对于运算符的重载，如果定义成成员函数最后会出现重载失去原有的意义，而重载是为了提高代码的可读性。

举个例子

void operator<<(ostream& out)

{

out << \_year << "-" << \_month << "-" << \_day << endl;

}

对于重载后的<<函数是可以实现输出的功能的，d<<cout;//d是对象 这样的重载与普通函数毫无差别。

那么可以在类外写重载函数，运算输出符作为全局的函数存在，再在对应的类中使用友元函数实现

同样以上面的例子

形如friend void operator<<(ostream &out,const date &d)

8、为什么有些运算符重载函数只能是成员函数？请整理并阐述，拒绝单纯的复制粘贴。

=,[],(),->只能通过成员函数来重载

当类中没有定义赋值运算符重载成员函数时, 当程序执行到某一赋值语句时，程序就会调用与赋值语句中右值类型匹配的构造函数，而把这右值当作此构造函数的实参。而当类中有定义赋值运算符重载成员函数，执行赋值语句时，程序就只会去调用相应的赋值运算符重载函数。

不能用友元函数实现这些运算符的重载，因为友元函数不是类内的成员函数，只是可以访问类内数据成员。所以在执行到对类对象的赋值语句时，就会有矛盾，因为类内没有相应的成员函数重载相应的运算符，系统自动调用构造函数，全局里又有相应的类型可以对应上的重载过的函数，刚好可以匹配上，系统自带的函数与重载后的运算符重载函数开始掐架。编译器歇逼

因此着几个运算符要写成成员函数的形式。

感觉78两题总结一下好像就是单目运算符要写成成员函数，双目运算符要写成友元函数。