

## 1. 按以下要求编写程序

现在有一个博物馆为了保证参观有序性和舒适度，需要通过闸机控制每日进场人数。



请按要求编写一个类 `CGateCount`，该类用于描述闸机的计数信息，具体要求如下：

1. 该类数据成员包含两种；第一，三个静态数据成员：当日人数上限(`m_maxCount`)，闸机总数 (`m_totalGate`)、合计通过人数 (`m_totalCount`,也就是所有闸机通过人数)，三个静态成员的初始值都是 0；第二，两个非静态数据成员：本闸机编号 (`m_gateNum`)、本闸机通过人数 (`m_count`)。  
**注意：1) 数据成员全部为私有；2) 全部为整型；**
2. 一个带 1 个参数 `count` 的构造函数，`count` 的默认值为 0，执行如下操作：本机编号等于闸机总数加 1，本机通过人数置 `count`，合计通过人数加 `count`，闸机总数加 1。
3. 1 个拷贝构造函数，拷贝构造时，执行如下操作：本机编号等于闸机总数加 1，本机通过人数置 0，闸机总数加 1。
4. 析构函数，用于将闸机总数减 1。
5. 一个静态成员函数 `set_maxCount` 用于设置当日人数上限。
6. 一个静态成员函数 `get_totalCount()`用于返回所有闸机通过人数。
7. 一个常成员函数 `get_count()`实现返回当前闸机通过人数。
8. 前自增运算符`++`重载，用于将当前闸机通过人数和所有闸机通过人数加 1。**注意：当所有通过人数达到上限值时不执行二者的加 1 操作。**
9. 后自增运算符`++`重载。注意事项同上。
10. 流输出运算，用于显示一个闸机的通过信息，具体显示格式参见运行结果图。
11. 使用给定的 `main` 函数测试 `Timer` 类的功能。

## 程序的 main 函数

**注意：**

1. 不得修改 `main` 函数！对 `main` 函数每修改一处，总分扣 2 分，最多扣 10 分。
2. 使用全局变量扣 10 分。

```

void main()
{
    CGateCount::set_maxCount(3000); //设置最大通过人数
    CGateCount gate1; //定义闸机对象
    gate1++;
    cout << gate1;
    CGateCount gate2(gate1); //定义闸机对象2
    (++gate2)++;
    cout << "gate2通过人数:" << gate2.get_count() << endl;
    cout << "合计通过人数:" << gate2.get_totalCount() << endl;

    CGateCount pGate3(2997);
    cout << "gate3通过人数:" << pGate3.get_count() << endl;
    cout << "合计通过人数:" << pGate3.get_totalCount() << endl;

    cout << ++gate2;
}

```

程序运行结果参见下图

```

C:\Windows\system32\cmd.exe
=====
闸机编号: 1
通过人数: 1
合计人数: 1
上限人数: 3000
=====
gate2通过人数: 2
合计通过人数: 3
gate3通过人数: 2997
合计通过人数: 3000
=====
闸机编号: 2
通过人数: 2
合计人数: 3000
上限人数: 3000
=====
请按任意键继续. . .

```

编程题评分标准

大项	子项	评分项	应得分	实得分
正确性	结果（65 分）	1. 类结构	5	
		2. 构造函数	5	
		3. 拷贝构造函数	5	
		4. 析构造函数	5	
		5. 静态成员函数 set_maxCount	5	
		6. 静态成员函数 get_totalCount()	5	
		7. 常成员函数 get_count()	5	
		8. 前自增运算符重载	10	
		9. 后自增运算符重载	10	
		10. 流输出运算符重载函数	10	
		程序运行异常中断、死循环或无任何结果	-10	
		main 函数修改 $n$ 处（最多扣 10 分）	$-2 \times n$	
		使用全局变量扣 10 分	-10	
可读性	缩进对齐（2 分）	缩进对齐规范	2	
		缩进对齐混乱或不缩进	0	
	变量命名规范（2 分）	命名规范	2	
		命名不规范	0	
	注释（1 分）	有注释	1	
		无注释	0	