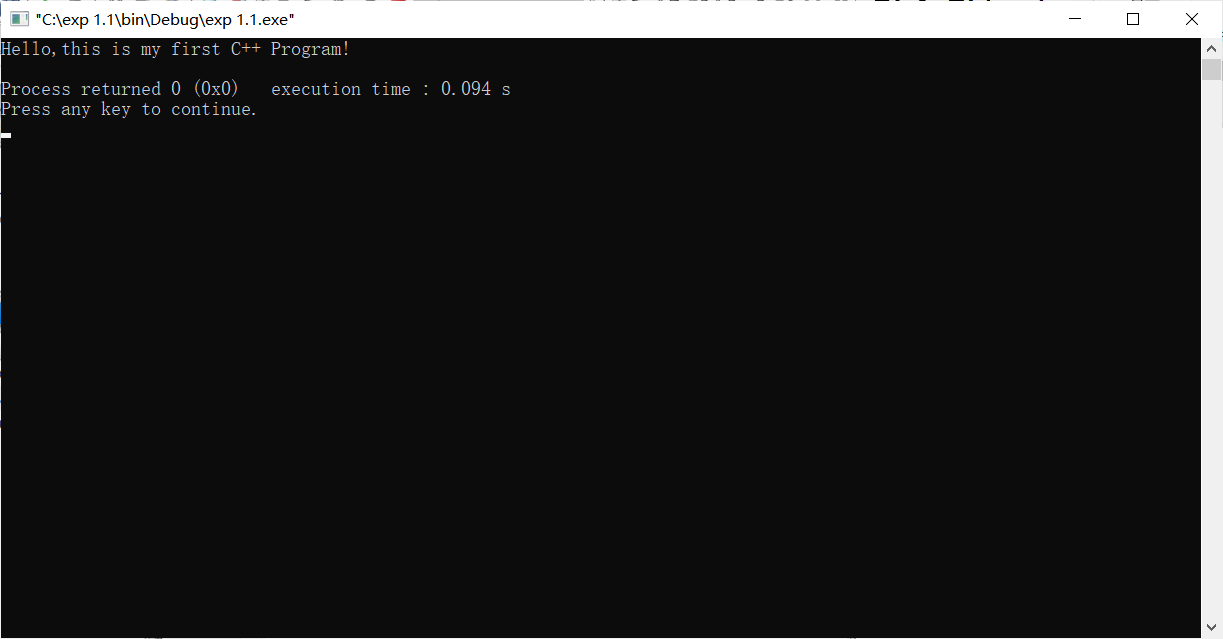
**C++实验一**

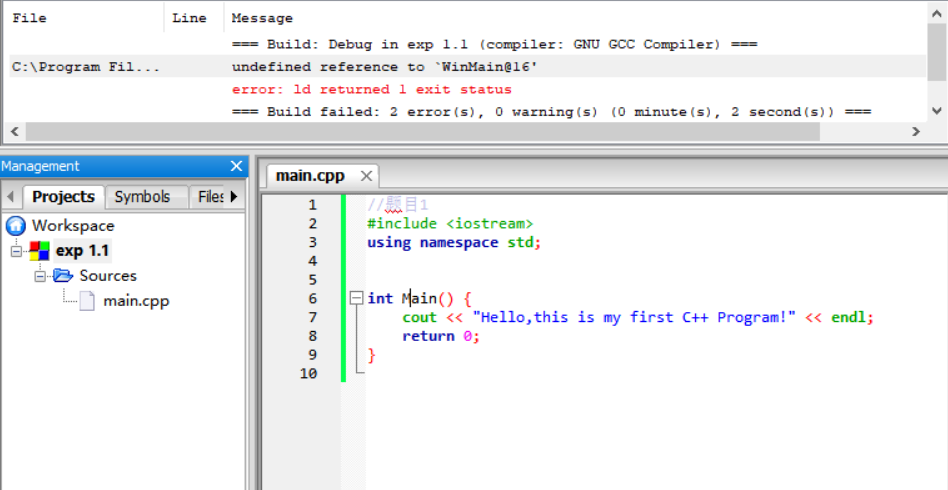
**2019计算机二班李嘉康**

注：对应的题目源码分别是文件夹里exp 1.1,exp 1.2........(1.1对应第一题

### 题目1 最简单的C++程序

运行截图：

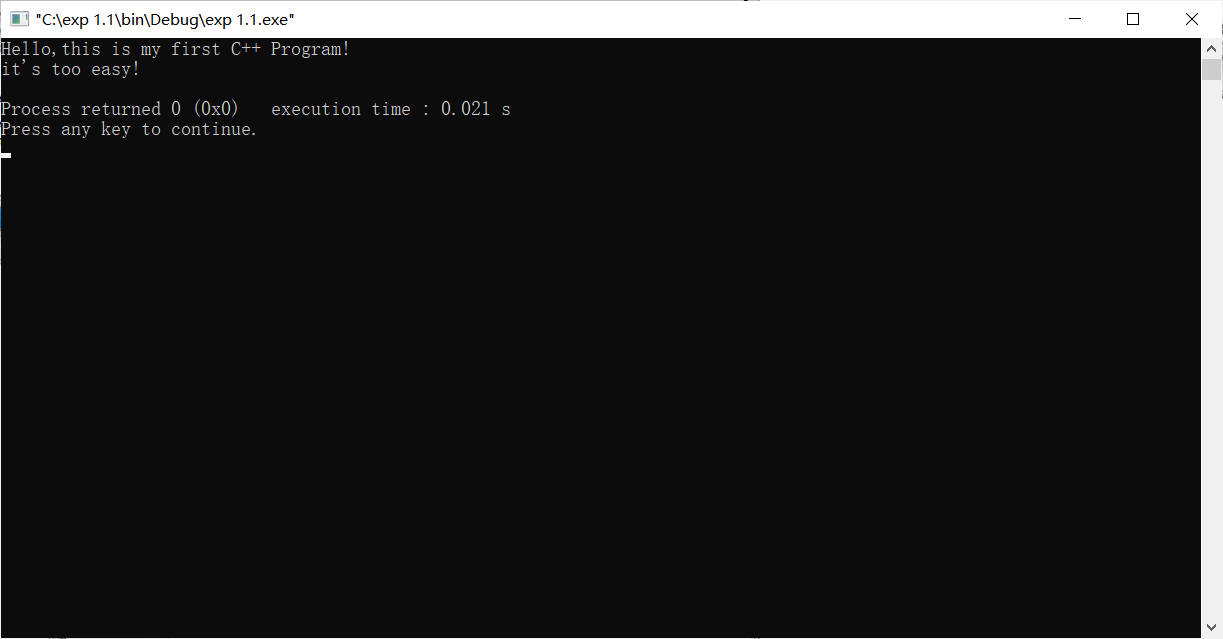
思考与扩展1：如果将main改成Main会报错如下：



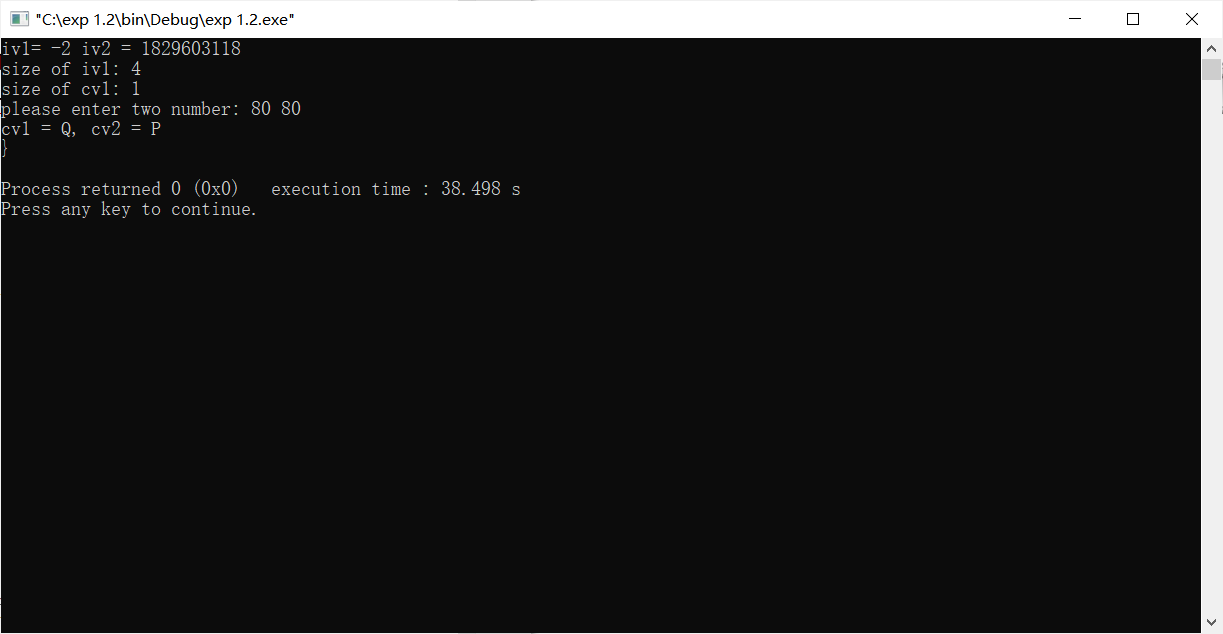
思考与扩展2：

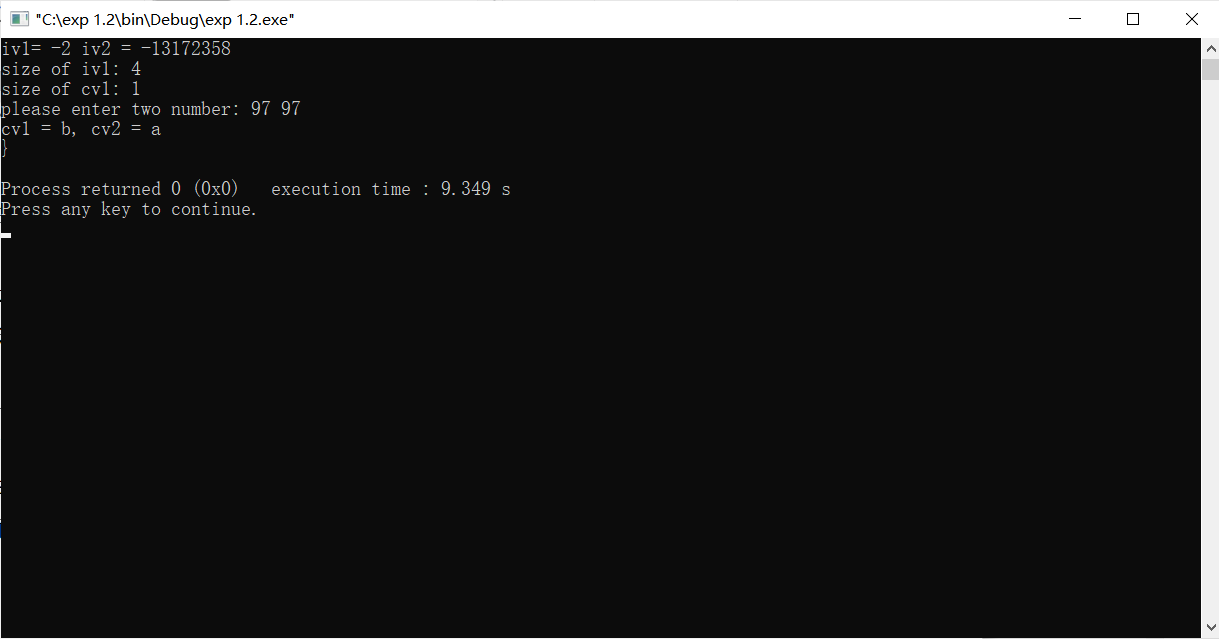
//如果想要再输出一行字符，将第7行改成

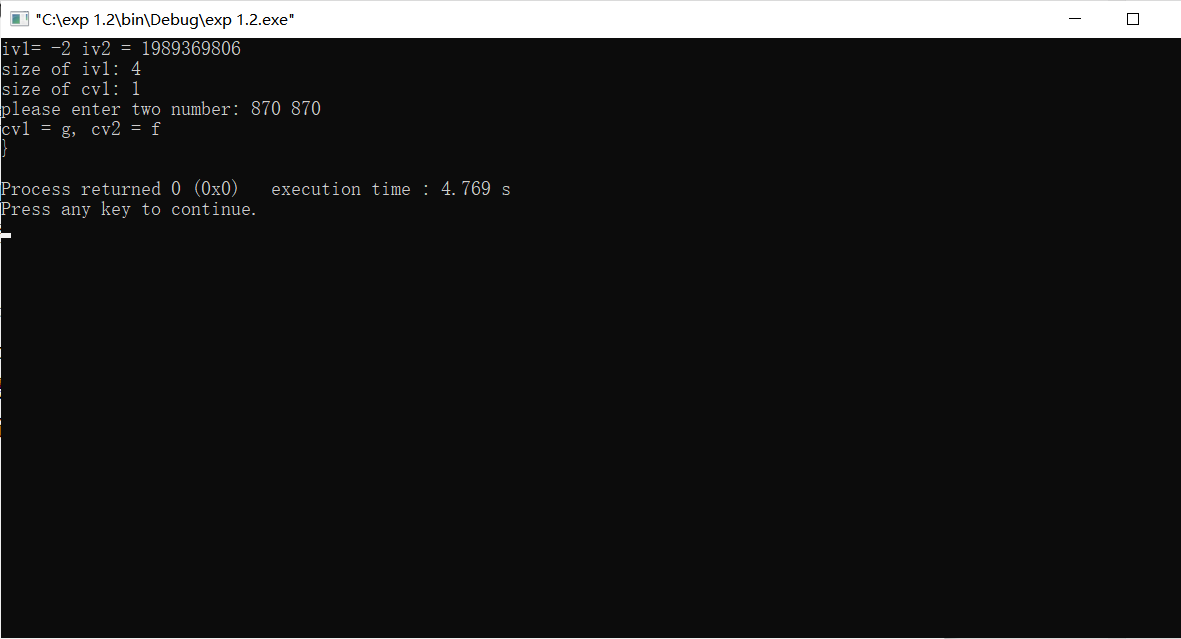
//"cout<< "Hello,this is my first C++ Program!"<<endl<<"it's too easy!”<<endl"

//即可

### 题目2 阅读并编译、运行程序

数据运行截图：

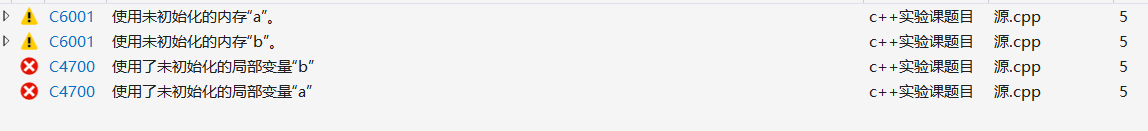




思考与扩展1：输出结果如上所示

（如果同样的代码换在visual studio下会编译不成功），出现这种结果的原因是没有对变量进行赋值就直接把变量的值输出，在codeblocks默认配置下会随机给这两变量赋值。

VS报错提示信息如下：



Codeblocks下警告如下：

警告信息其实不会影响编译。

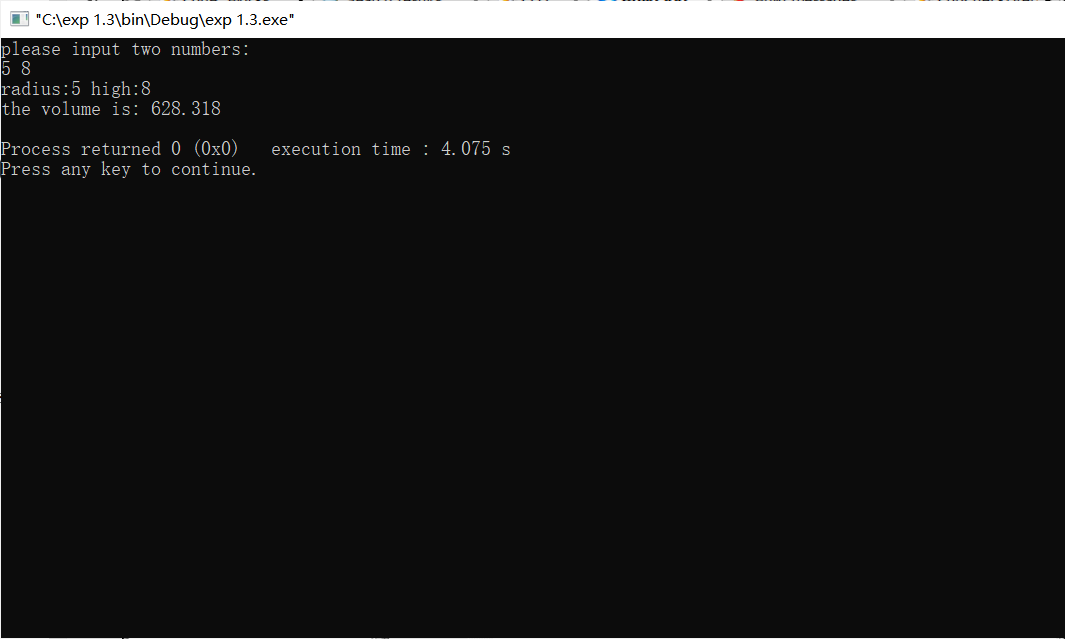
思考与扩展2：

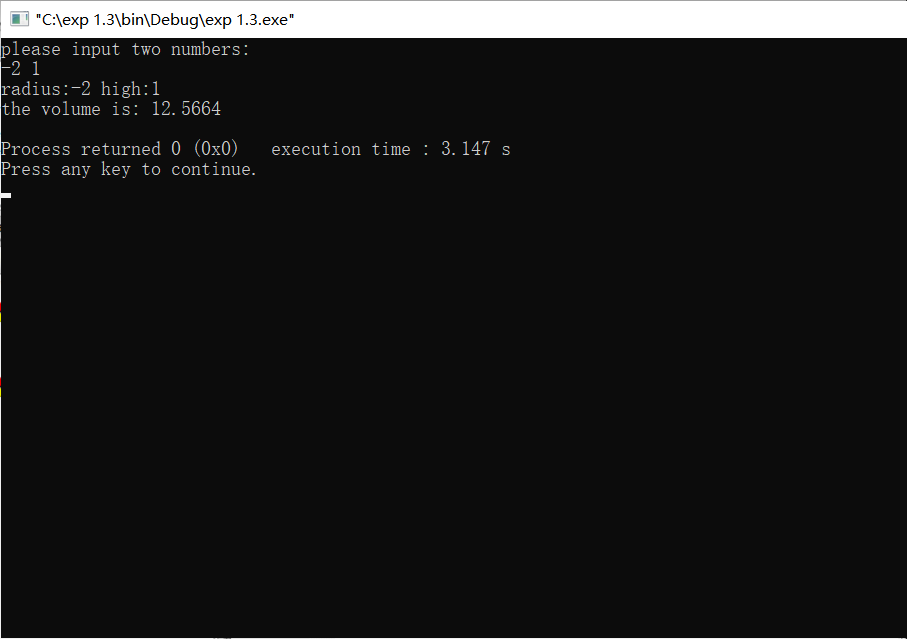
结果为8

思考与扩展3：其实程序输出结果和我预期的一样：

譬如sizeof是很好想的，++c和c++式值不同（++c）>(c++),然后数字是可以赋值给字符型数据并已ASCII码对应的字符打印出来。

### 题目3 简单应用—求圆柱体的体积

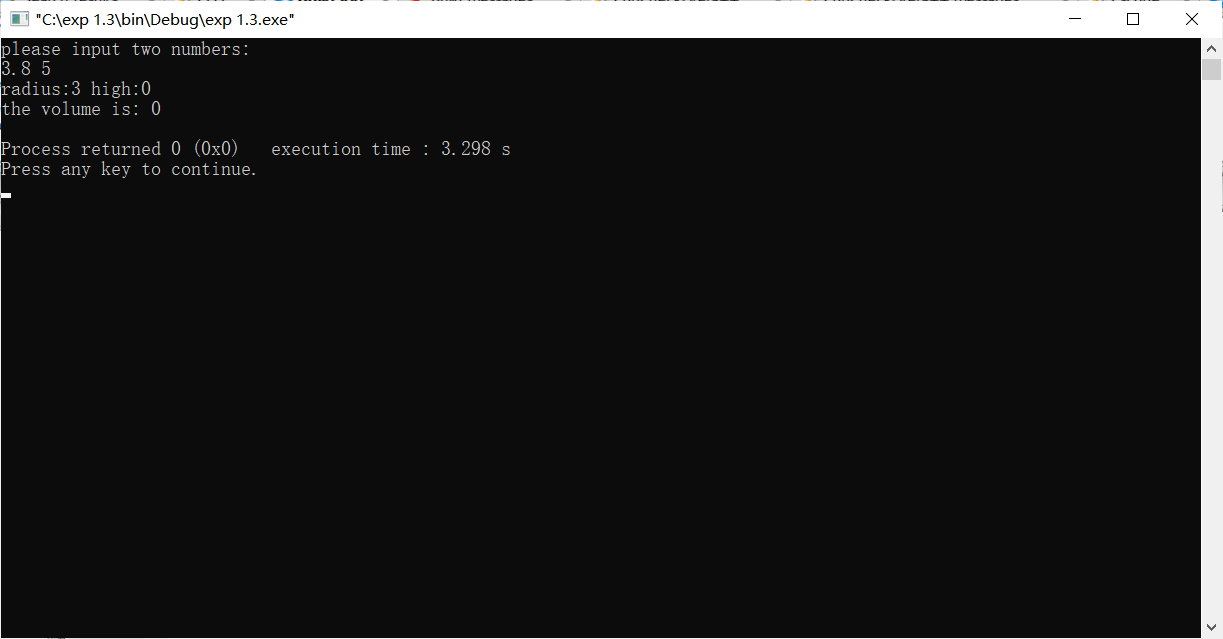
运行截图：



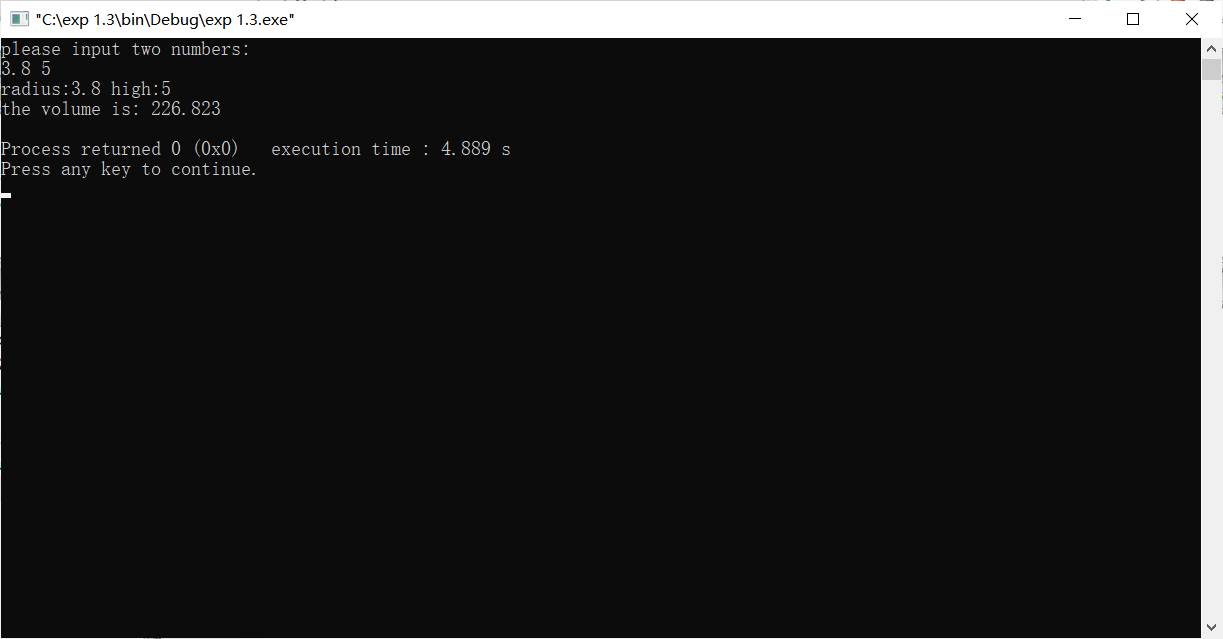
思考与扩展1：

把define那里去掉，在主函数里加上const double PI=3.14159

思考与扩展2：



显然结果不正确，改为double后：



原因：系统执行到cin>>radius时，此时键盘缓冲区啥也没有，会等待我们的输入。然后输3.8时，3.8进入缓冲区，cin开始读取，由于cin函数想要获得的是一个整数，所以它会读到3，后一个字符是.不属于整数，cin便停止，但不属于失败。再读到>>high时，缓冲区残留的是.8，其并不是一个有效的整数，此时cin会出错，将该变量初始化为0。

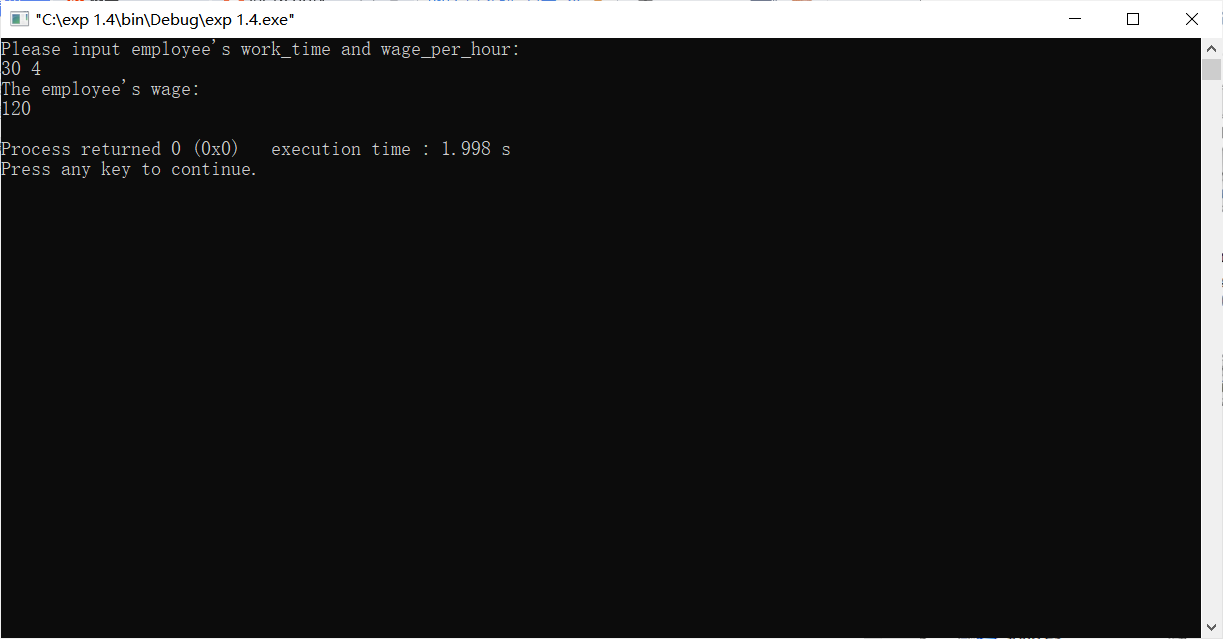
思考与扩展3：

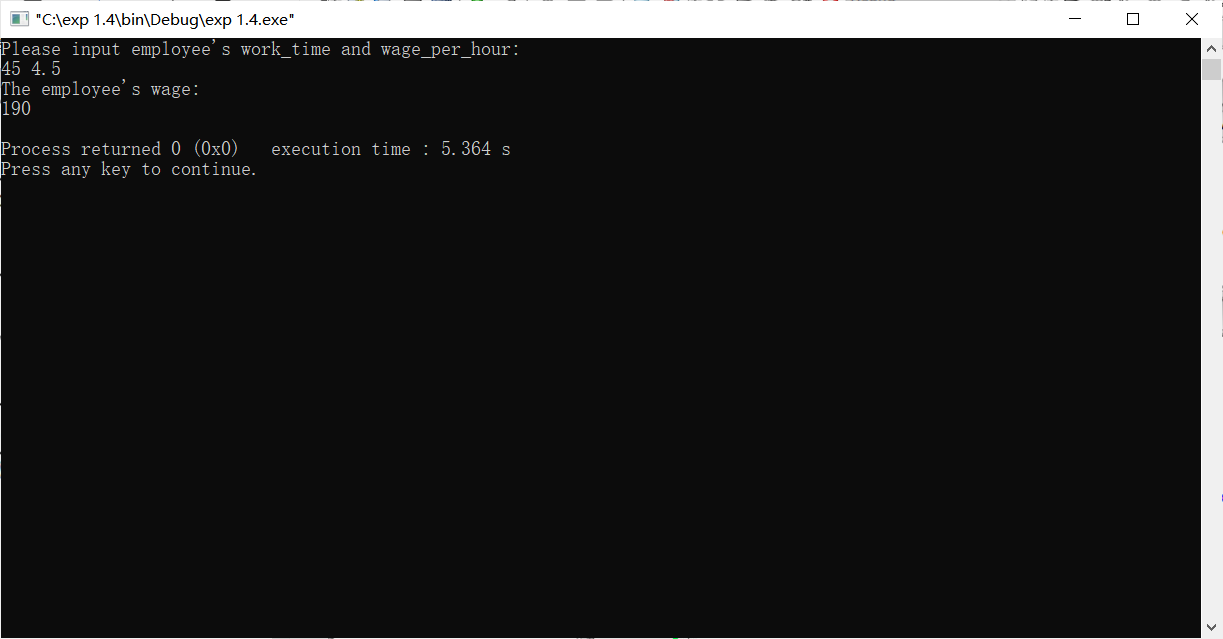
可以直接将“please input two numbers:”改成”please input two values for radius and high respectively:”等等。

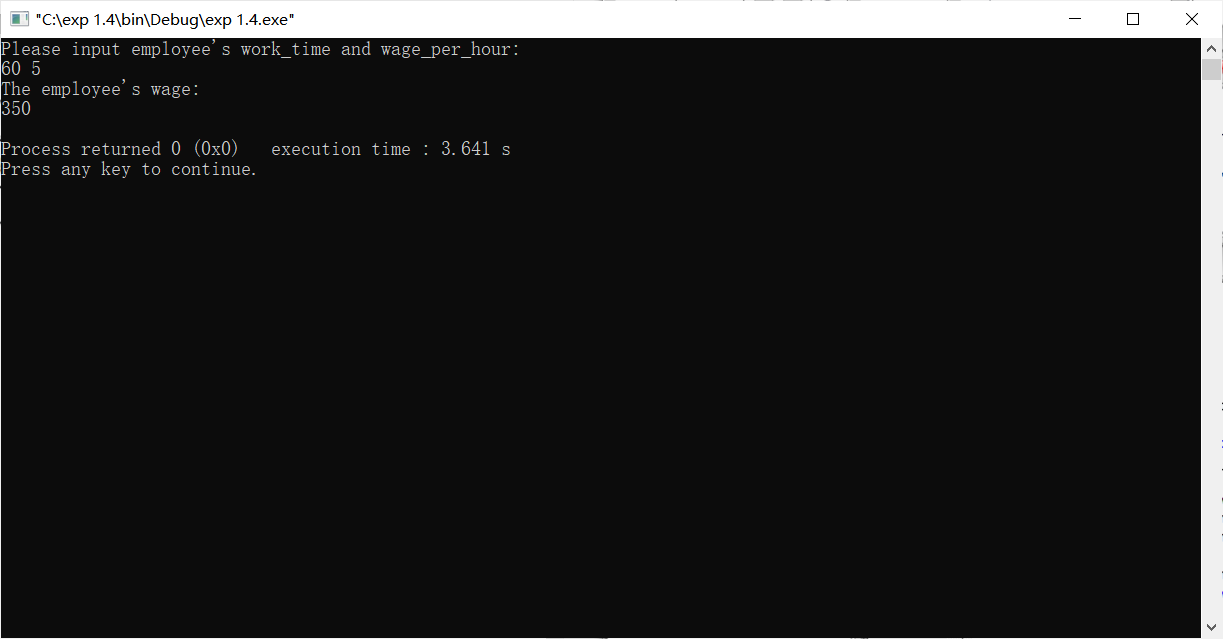
思考与扩展4：

该程序设计了4个类，分别是container，cube，sphere，cylinder，后面三个皆继承了container，然后然后在公有里定义了表面积，和体积的函数。

### 题目4 计算公司员工周工资

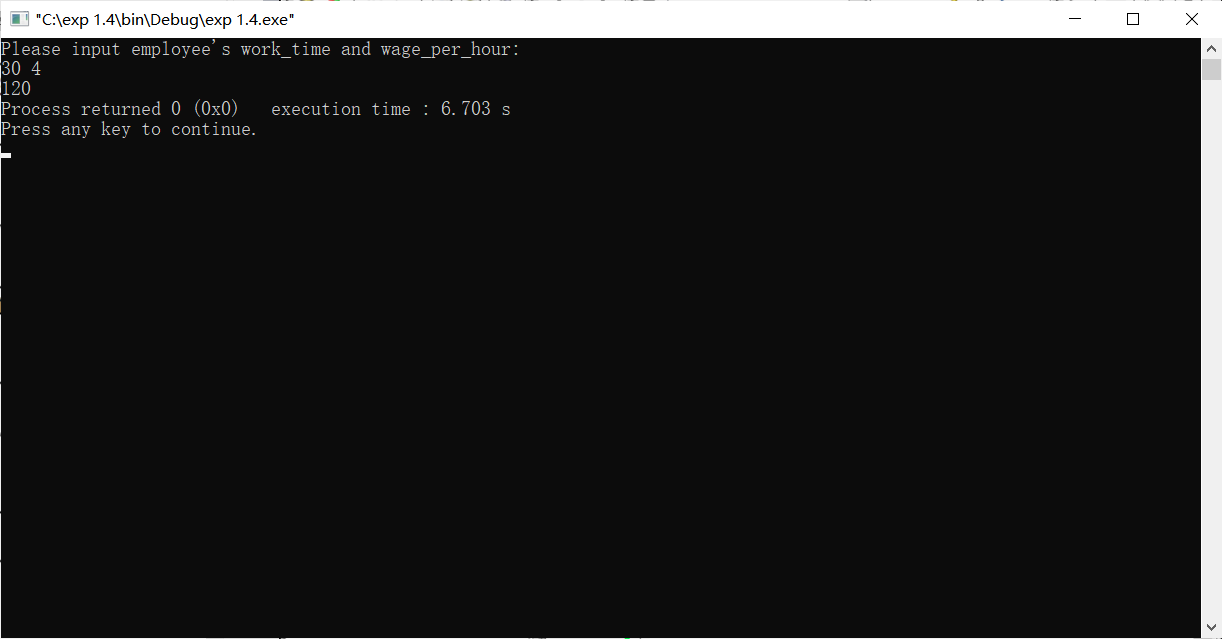
运行截图：

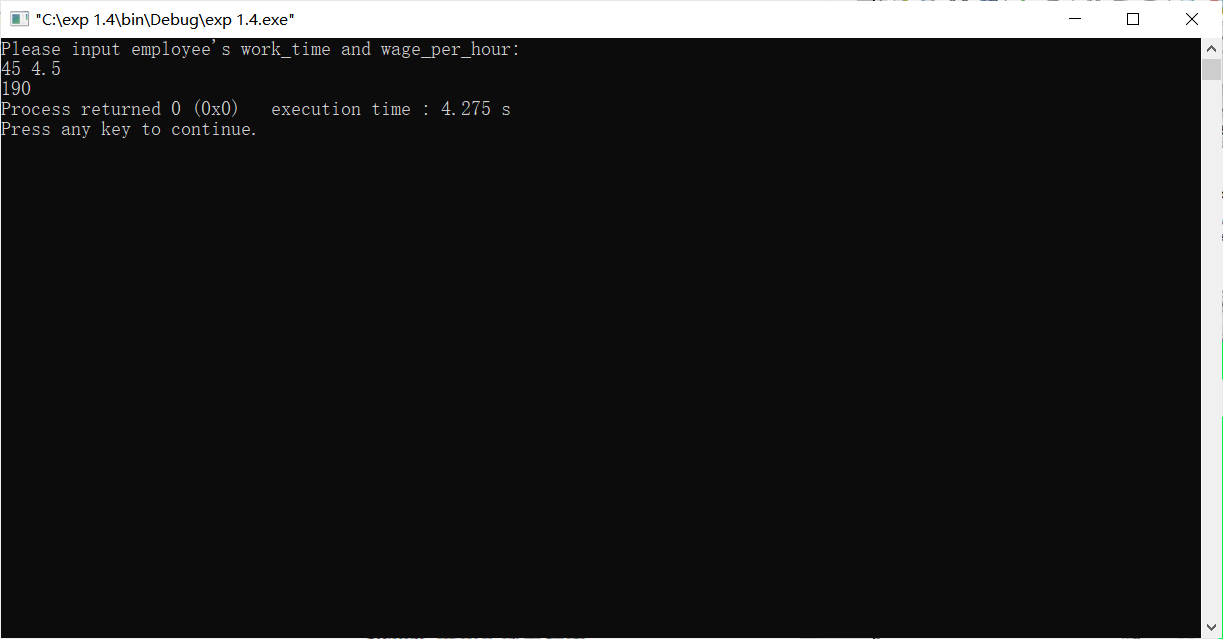


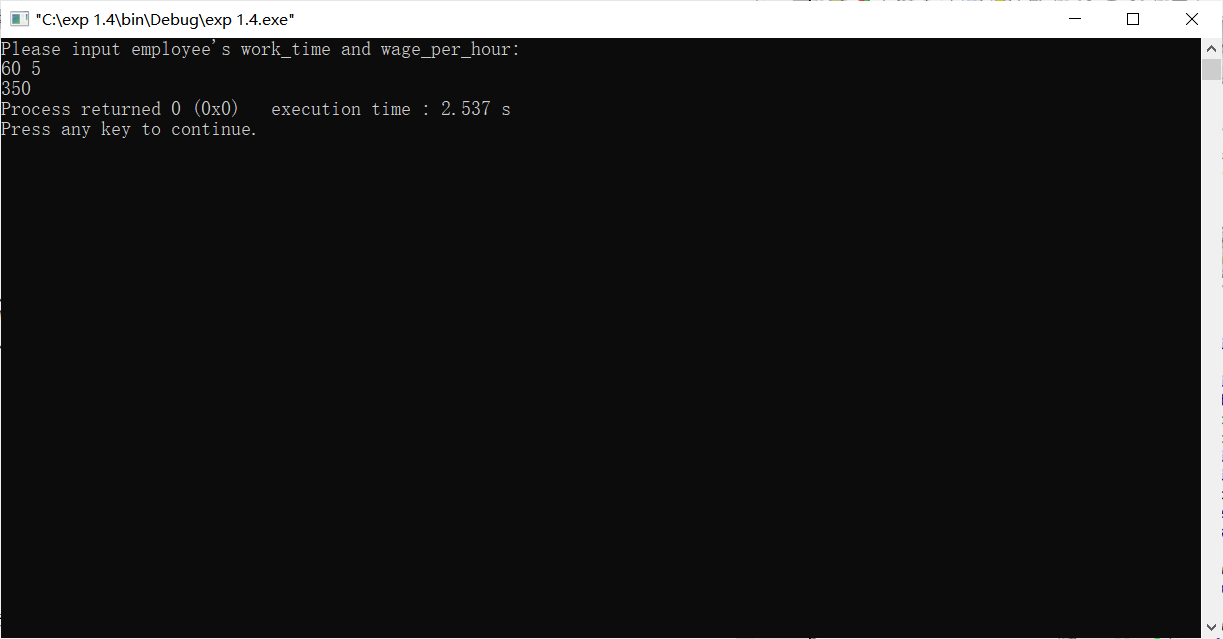


思考与扩展1：

用switch后的运行结果（把之前的用if的语句注释掉了）：







思考与扩展2：

原主函数内外围加入一个:

Int control;

Cout << “type any non-zero number to continue,type 0 to end”;

Cin >> control;

While(control){

Cout << “type any non-zero number to continue,type 0 to end”;

Cin >> control

.........(与原来的一样)

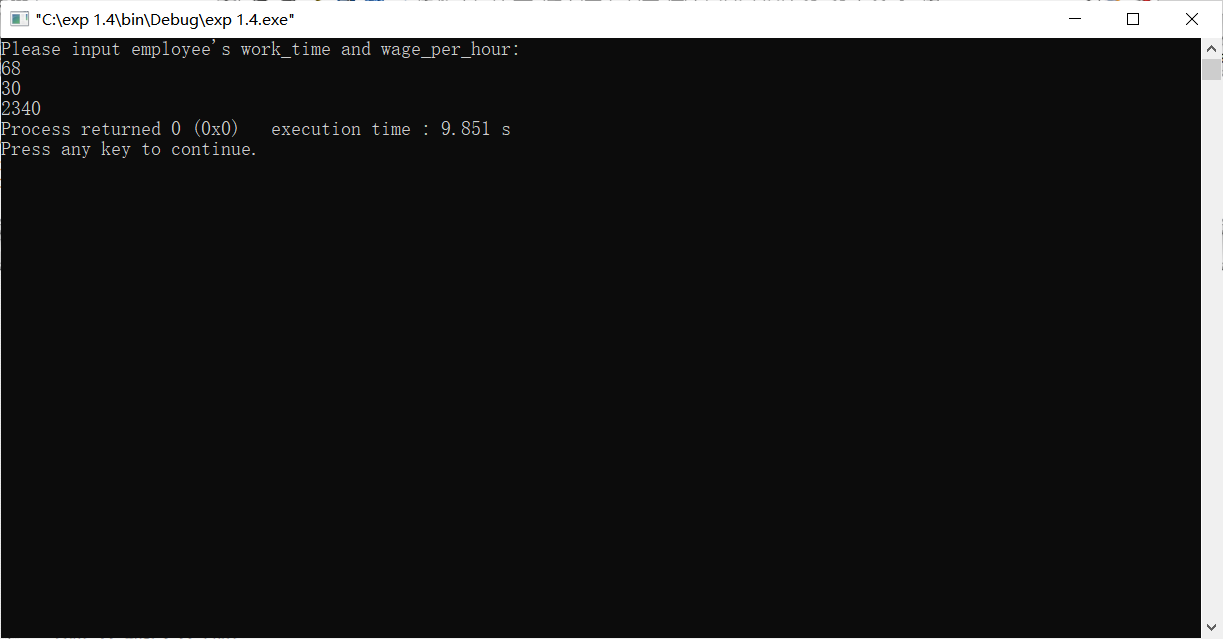
}

//因为这里要注释的东西太多就不附上截图了。

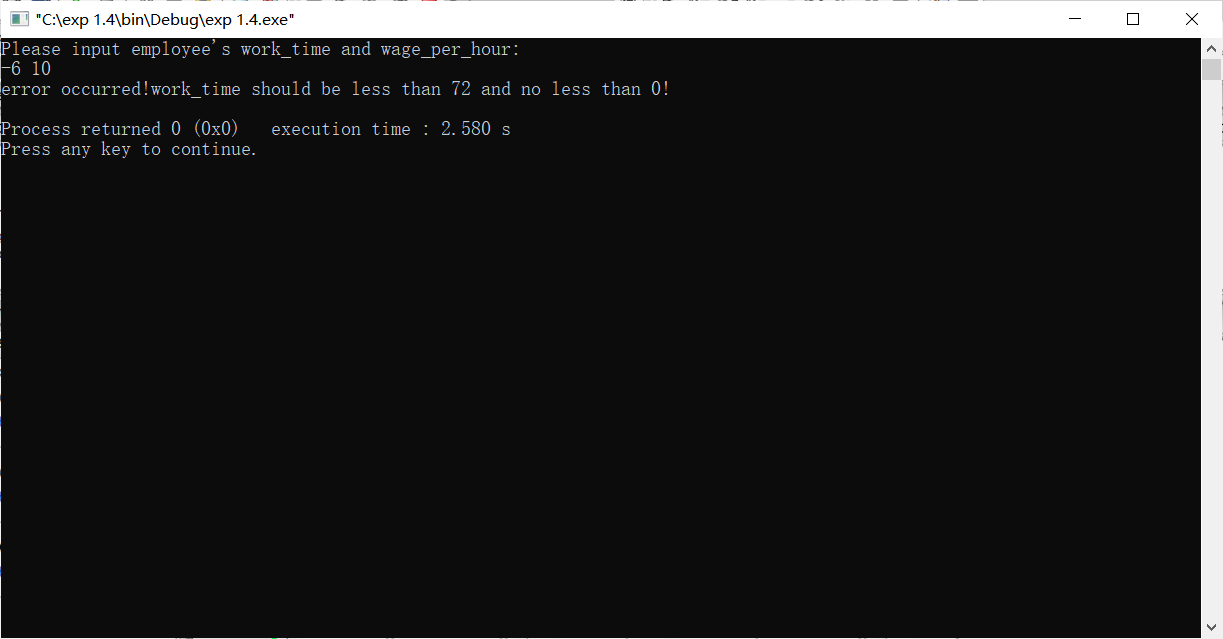
Return 0;

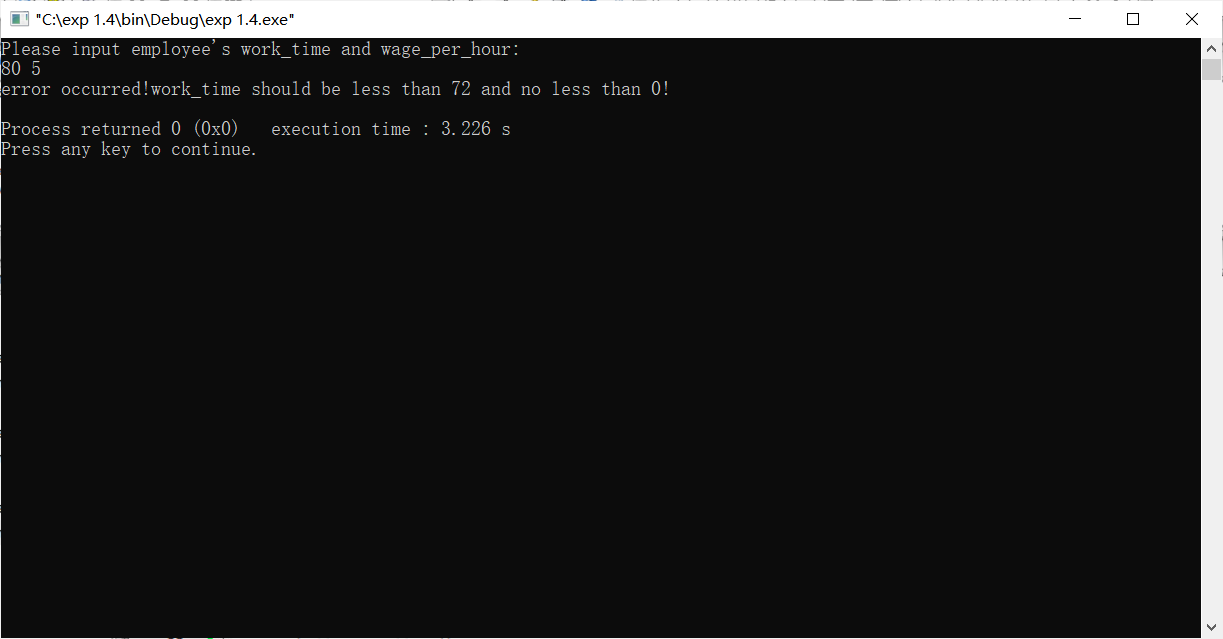
思考与扩展3：

附上>60&&<70的数据：



附上不合理数据：





### 题目5 设计一个计算+、-、\*、/的程序

测试样例：

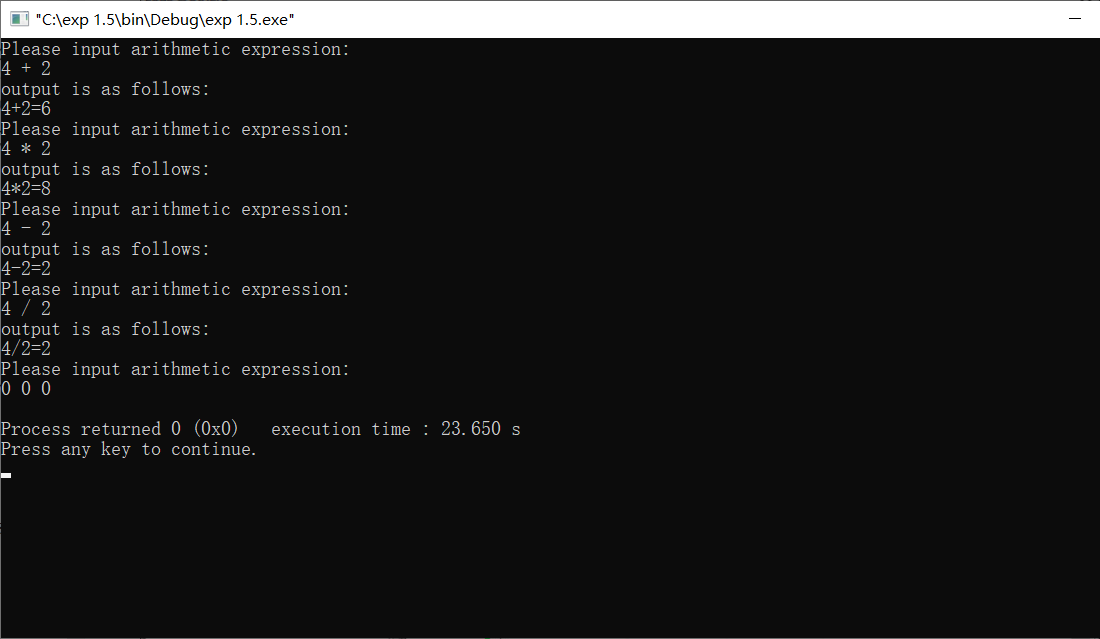
4 + 2；

4 - 2;

4 \* 2;

4 / 2;

0 0 0;



思考与扩展1：

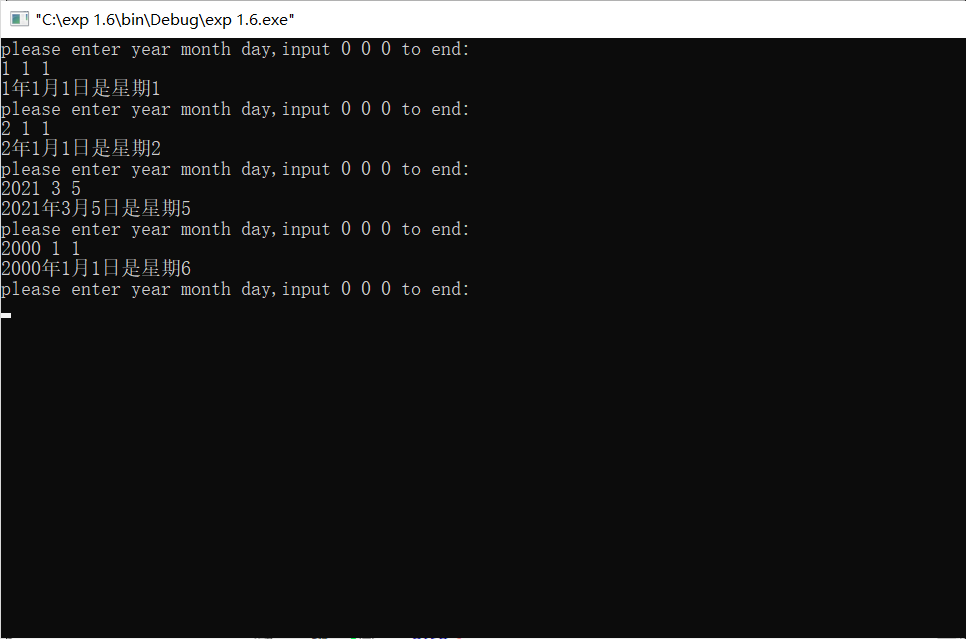
一方面可以作为数据输入，也可以同时作为while循环的判定条件来结束循环。

思考与扩展2：

Break 语句在这里的作用是只执行case中的一个，而不是多个（可以节约时间）

### 题目6 计算生日是星期几

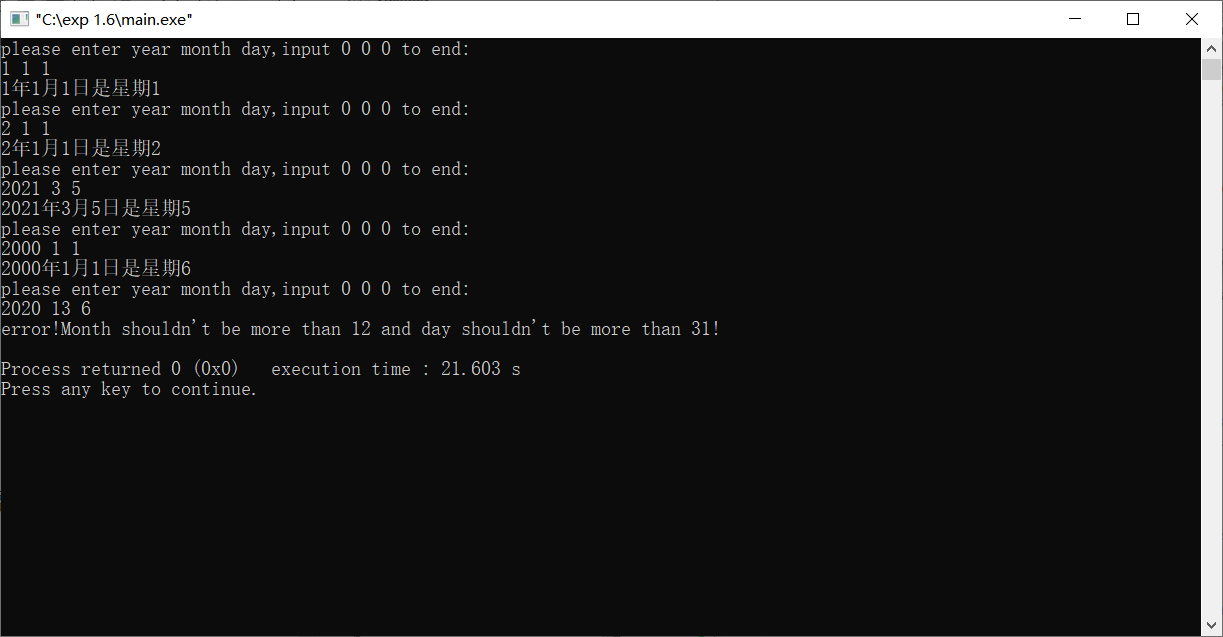
截图如下：



思考与扩展1,2：

代码如exp 1.6中所示：

运行截图：



扩展2意思即边缘条件考虑即可，因此设置一个特判即可，例子如上图所示。