已知1900.1.1是星期一，对于一个给定的年份和月份，输出这个月的最后一天是星期几

a) 1900.1.1是星期一

b) 非闰年的2月是28天，闰年是29天

c) 闰年定义：公元年数能被4整除且又不能被100整除是闰年；能直接被400整除也是闰年

输入格式:

两个整数，分别代表年份和月份

输出格式：

星期数：0代表星期日，1代表星期1，…，6代表星期6

1、编写函数实现功能

2、编写测试用例

*1、*

*#!/usr/bin/env python  
# encoding: utf-8  
"""  
@author: Hank zhang  
@contract:  
@file: homework4\_ex.py  
@time: 2016/10/23 11:34  
"""*  
*#非闰年 每月天数*monthdays = {1:31, 2:28, 3:31, 4:30, 5:31, 6:30, 7:31, 8:31, 9:30, 10:31, 11:30, 12:31}  
*#非闰年与闰年对应的天数*yeardays = {0:365, 1:366}  
  
*#计算是否是闰年***def** leapYear(year):  
 **assert** year>0, **u"输入年份必须大于零的整数"  
 if** (year%4==0 **and** year%100!=0) **or** (year%400==0):  
 **return** 1  
 **return** 0  
  
*#计算星期几***def** sumWeekday(days):  
 *# assert type(days) == type(int), u"输入天数必须大于零的整数"*

*#星期*week = {-6:**u'星期一'**, -5:**u'星期二'**, -4:**u'星期三'**, -3:**u'星期四'**, -2:**u'星期五'**, -1:**u'星期六'**,0:**u'星期日'**,\  
 1:**u'星期一'**, 2:**u'星期二'**, 3:**u'星期三'**, 4:**u'星期四'**, 5:**u'星期五'**, 6:**u'星期六'**}**return** week[days%7]  
  
*#计算与1990年1月1日相距的时间***def** sumDays(year, month):  
 **assert** year>0 **and** month>0, **u"输入年月份必须大于零的整数"** days = 0  
 **if** year >= 1990:  
 **for** elem **in** range(1990, year):  
 days += yeardays[leapYear(elem)]  
 **for** elem **in** range(1, month+1):  
 **if** leapYear(year)==1 **and** elem ==2:  
 days += monthdays[elem] + 1  
 **else**:  
 days += monthdays[elem]  
 **else**:  
 **for** elem **in** range(year, 1990):  
 days -= yeardays[leapYear(elem)]  
 **for** elem **in** range(1, month+1):  
 **if** leapYear(year)==1 **and** elem ==2:  
 days += monthdays[elem] + 1  
 **else**:  
 days += monthdays[elem]  
 **return** days  
  
**if** \_\_name\_\_ == **"\_\_main\_\_"**:  
 year = input(**"请输入年份："**)  
 month = input(**"请输入月份："**)  
 days = sumDays(year, month)  
 **print** sumWeekday(days)

2、

**import** unittest  
  
**from** src.homework4\_ex **import** sumWeekday  
  
  
**class** MyTestCase(unittest.TestCase):  
 **def** test\_something(self):  
 test\_dict = {(2016, 10): **u'星期一'**, (2016, 9): **u'星期五'**, (2016, 8): **u'星期三'**, (2016, 7): **u'星期日'**,  
 (1988, 10): **u'星期一'**, (1988, 9): **u'星期五'**, (1988, 8): **u'星期三'**, (1988, 7): **u'星期日'**}  
 keys = test\_dict.viewkeys()  
 keyslist = list(keys)  
 self.assertEqual(sumWeekday(2016, 10), test\_dict[keyslist[0]])  
 self.assertEqual(sumWeekday(2016, 9), test\_dict[keyslist[1]])  
 self.assertEqual(sumWeekday(2016, 8), test\_dict[keyslist[2]])  
 self.assertEqual(sumWeekday(2016, 7), test\_dict[keyslist[3]])  
 self.assertEqual(sumWeekday(1988, 10), test\_dict[keyslist[4]])  
 self.assertEqual(sumWeekday(1988, 9), test\_dict[keyslist[5]])  
 self.assertEqual(sumWeekday(1988, 8), test\_dict[keyslist[6]])  
 self.assertEqual(sumWeekday(1988, 7), test\_dict[keyslist[7]])  
  
**if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 unittest.main()