计算日期之差

1. 题目

输入两个日期，求两个日期之间的天数，如果两个日期是连续的，则规定他们之间的天数为两天。

输入格式：

YYYYMMDD

20130101

20130505

输出个数：

5

1. 思路

要计算日期之间的天数，我们可以想到从小的日期开始不断加1天，直到小的日期等于大的日期为止。其中，当天数d等于当前的月份m的天数+1时，月份m就要+1，且天数d置为1；当月份m变为13时，则年份y就要+1，同时月份m置为1。

注意，闰年的2月份天数为29天，平年的2月份天数为28天。

闰年的定义：

闰年是公历中的名词。闰年分为普通闰年和世纪闰年。

普通闰年:公历年份是4的倍数的，且不是100的倍数，为普通闰年。（如2004、2020年就是闰年）；

世纪闰年:公历年份是整百数的，必须是400的倍数才是世纪闰年（如1900年不是世纪闰年，2000年是世纪闰年）；

闰年(Leap Year)是为了弥补因人为历法规定造成的年度天数与地球实际公转周期的时间差而设立的。补上时间差的年份为闰年。闰年共有366天（1-12月分别为31天，29天，31天，30天，31天，30天，31天，31天，30天，31天，30天，31天）。

设置二维数组 int month[13][2]存储闰年和平年的月份天数

1. 代码

#define \_CRT\_SECURE\_NO\_WARNINGS

#include<iostream>

#include<cstdio>

using namespace std;

int month[13][2] = {

{0,0},{31,31},{28,29}, {31,31}, {30,30}, {31,31}, {30,30},

{31,31},{31,31},{30,30},{31,31}, {30,30}, {31,31}

};

//判断是否是闰年

bool isLeap(int year)

{

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

}

int main()

{

int time1, y1, m1, d1;

int time2, y2, m2, d2;

while ( scanf("%d%d",&time1, &time2)!= EOF)

{

//如果第一个日期比第二个日期大，则交换

if (time1 > time2)

{

int temp = time1;

time1 = time2;

time2 = temp;

}

y1 = time1 / 10000;

m1 = time1 % 10000 / 100;

d1 = time1 % 100;

y2 = time2 / 10000;

m2 = time2 % 10000 / 100;

d2 = time2 % 100;

int ans = 1; //记录天数

// 不断加1，直到两个日期相等

while (y1 < y2 || m1 < m2 || d1 < d2)

{

d1++;

//天数满了当前月份的天数+1

if (d1 == month[m1][isLeap(y1)] + 1)

{

m1++;

d1 = 1;

}

//月份满了当前年份的月份数

if (m1 == 13)

{

y1++;

m1 = 1;

}

ans++; //记录天数+1

}

cout << ans << endl;

}

return 0;

}