//数据检验的方法：

//对全部数据的覆盖度（数据类型、数据类型下对应的所有范围）

// 1）超出输入要求的数据 进行合法性检验（可使用一些库函数e.g. isnumeric），排除

// 2）对范围内的数据的通性，即对所有数据都符合题目输入输出要求

// 1>输入数据从最小到最大轮次思考

// 2>反过来利用输出数据进行检验

例子：

//题目内容：

//

//UTC是世界协调时，BJT是北京时间，UTC时间相当于BJT减去8。现在，你的程序要读入一个整数，表示BJT的时和分。整数的个位和十位表示分，百位和千位表示小时。如果小时小于10，则没有千位部分；如果小时是0，则没有百位部分；如果小时不是0而分小于10分，需要保留十位上的0；如果小时是0而分小于10分的，则不需要保留十位上的0。如1124表示11点24分，而905表示9点5分，36表示0点36分，7表示0点7分。

//

//有效的输入范围是0到2359，即你的程序不可能从测试服务器读到0到2359以外的输入数据。

//

//你的程序要输出这个时间对应的UTC时间，输出的格式和输入的相同，即输出一个整数，表示UTC的时和分。整数的个位和十位表示分，百位和千位表示小时。如果小时小于10，则没有千位部分；如果小时是0，则没有百位部分；如果小时不是0而分小于10分，需要保留十位上的0；如果小时是0而分小于10分的，则不需要保留十位上的0。

//

//提醒：要小心跨日的换算。

#include <stdio.h>

int main ()

{

int min1, hour1, min2, hour2;

int in;

int sum1, sum2;

scanf("%d",&in);

if(in > 2359 || in < 0)//数据合法性检验

{

return 0;

}

min1 = in % 100;

hour1 = in / 100;

sum1 = hour1 \*60 + min1;

if (sum1- 8\*60 > 0)

{

sum2 = sum1- 8\*60;

}

else

{

sum2 = 24\*60 + sum1- 8\*60; //跨日的计算

}

min2 = sum2 % 60;

hour2 = sum2 / 60;

if (hour2 != 0) //抓住问题产生分类的关键

{

printf("%d",hour2);

printf("%02d",min2);

}

else

{

printf("%d",min2);

}

return 0;

}

// 在此问题中， 1）有效的输入范围是0到2359

// 2） 1>1、12、102、1102

// 2>涉及到 跨日的换算