**★★☆☆☆**

**題組：Problem Set Archive** **with Online Judge**

**題號：10339: Watching Watches**

**解題者：許育菖**

**解題日期：2024年4月25日**

**題意：**

有兩隻指針式手錶，其中一隻每天會慢n秒，另一隻每天會慢m秒。

試問：當兩隻手錶同時從12:00開始跑時，下一次「兩隻手錶顯示的時間完全相同」是甚麼時候？

**題意範例：**

n、m單位：sec

|  |  |  |
| --- | --- | --- |
| **第一隻手錶慢的時間 n** | **第二隻手錶慢的時間 m** | **output** |
| 1 | 2 | 12:00 |
| 0 | 7 | 10:17 |

**解法：**

依題目邏輯直接計算。

首先，我們需要先知道零誤差的手錶指針跑一圈所花費的秒數：

 *cycle = 60 × 60 × 12*

接著從題目的定義，我們可以推得：兩錶時間再次重合，即代表跑得比較快的手錶倒追了一圈。

*∴ 走過的天數 = cycle / (*$\left|n-m\right|$*)*

*⟹ 走過的秒數 = 走過的天數 × (24 × 60 × 60* ***- n****)*

因為我們以第一支手錶來做計算，此手錶每天會慢n秒，所以一天的秒數需要-n；如果是以第二支手錶做計算則一天的秒數需要-m。

*走過的分鐘數 = 走過的秒數 / 60*

當我們計算出「走過的分鐘數」後，注意此時需要去判斷「走過的秒數 % 60」的值是否大於等於30，因為整數除法是無條件捨去，如果剩餘的秒數大於等於30的話，則「走過的分鐘數」需要 +1。

最後再由「走過的分鐘數」來算出兩錶重合的準確時間。

※注意

1. 根據題目要求，沒有0點。所以如果計算後的時間為的0點多，小時的部分需輸出為12。
2. 因為此題的輸出格式特殊，使用printf()會使程式碼比起使用cout還要來的簡潔。

**解法範例：**

無。

**討論：**

無。

**程式：**

/\*

 B113040045 許育菖

 2024/04/25

 This program is to know when the two clock show the same time again.

\*/

#include <iostream>

#include <cmath>

#include <cstdlib>

using namespace std;

int main()

{

 // clock

 int n, m;

 const double cycle = 60 \* 60 \* 12 \* 1.0; // 時鐘轉一圈需要的秒數

 while(!cin.eof() && cin >> n >> m)

 {

 if(n == 0 && m == 0)

 {

 cout << n << " " << m << " " << "12" << ':' << "00" << '\n';

 continue;

 }

 /\*

 走過的天數 = cycle / difference

 走過的秒數 = 走過的天數 \* (24 \* 60 \* 60 - n)，因為每天會慢n秒，所以需要-n or -m

 走過的分鐘數 = 走過的秒數 / 60

 \*/

 long long int second = (long long int)( ( cycle / abs(n - m) \* (2 \* cycle \* 1.0 - n)) );

 int min = second / 60;

 // 因為做整數除法時，小數點後面會無條件捨去

 // 所以需判斷不足一分鐘的秒數是否進位

 // 當剩餘秒數 >= 30秒時，需做進位

 if(second % 60 >= 30)

 min += 1;

 int hour = (min / 60) % 12;

 if(hour == 0)

 hour = 12;

 min %= 60;

 // output

 printf("%d %d %02d:%02d\n", n, m, hour, min);

 }

 return 0;

}