Project: CS1105

Submitted by:

**Shubham Sharma(2020BTechCSE072)**

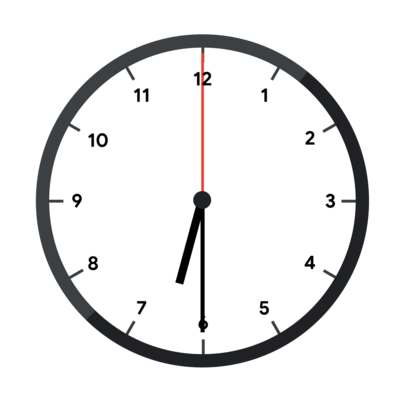
**Siddharth Jangid(2020BTechCSE074)**

**Problem Source:** <https://codingcompetitions.withgoogle.com/codejam/round/0000000000435baf/00000000007ae694>

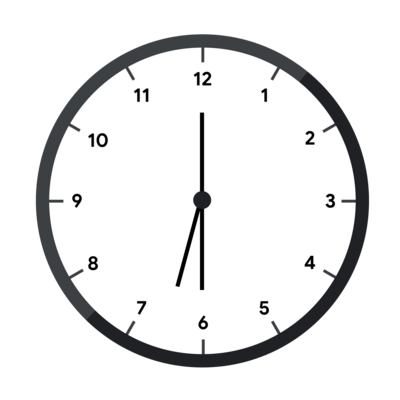
**Problem Statement:**

Emmett found an old clock in his attic. The clock is a circle with 3 hands that attach to the center and rotate clockwise at constant speeds. They are called the *hours hand*, the *minutes hand* and the *seconds hand*. At midnight, all hands point up. The hours hand completes a full revolution in 1212 hours, the minutes hand completes a full revolution in 11 hour, and the seconds hand completes a full revolution in 11 minute. 11 hour is equal to 6060 minutes, 11 minute is equal to 6060 seconds, and 11 second is equal to 109109 nanoseconds.

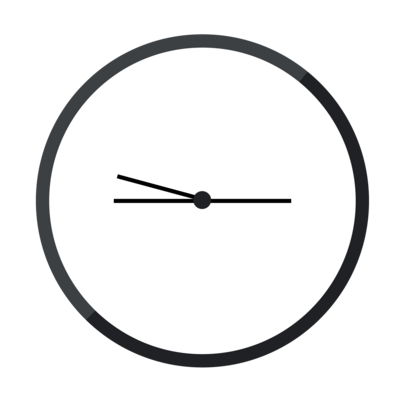
For example, the clock depicted below is showing that the time is exactly 66 hours and 3030 minutes after midnight. The hours hand (short black) is halfway between 66 and 77 (completed 6.5/126.5/12 of a revolution), the minutes hand (long black) is pointing straight down because it has completed exactly 66 and a half full revolutions and the seconds hand (red) is pointing straight up because it has completed an integer number of full revolutions.



Unfortunately, the hands are broken, so they all look identical and there is no way to know which hand is which. The clock in the picture above, with its hands broken, would look like this.



In addition, no markings remain that allow Emmett to know which way is up, so any rotation of the clock could be the correct one (the clock can only be rotated, not reflected). To continue with our example, the fully broken clock could look like this.



Emmett does know that the time was strictly before noon, that is, strictly less than 1212 hours after midnight. Emmett has taken a picture of the clock. Given that picture (represented by the angles of the hands relative to a single arbitrary axis), figure out what time it could correspond to.

Notice that Emmett has already figured out a viable orientation of the clock in some cases (Test Set 1) and has managed to narrow down the possible times to a whole integer number of seconds (Test Sets 1 and 2) or nanoseconds (Test Set 3). Please see the limits sections for more details.

### Input

The first line of the input gives the number of test cases, TT. TT lines follow. Each line describes a test case and contains three integers AA, BB, and CC: the angles of each hand, relative to an arbitrary axis and given in ticks in the clockwise direction. 11 *tick* is equal to 1/12×10−101/12×10−10 degrees. This means that the hours hand rotates exactly 11 tick each nanosecond, the minutes hand rotates exactly 1212 ticks each nanosecond and the seconds hand rotates exactly 720720 ticks each nanosecond.

### Output

For each test case, output one line containing Case #xx: hh mm ss nn, where xx is the test case number (starting from 1) and hh, mm, ss, and nn are integers: hh is the number of full hours since midnight (between 00 and 1111, inclusive), mm is the number of full minutes since the last full hour (between 00 and 5959, inclusive), ss is the number of full seconds since the last full minute (between 00 and 5959, inclusive) and nn is the number of full nanoseconds since the last full second (between 00 and 109−1109−1, inclusive).

### Limits

Time limit: 30 seconds.  
Memory limit: 1 GB.  
1≤T≤1001≤T≤100.  
0≤A≤B≤C<360×12×10100≤A≤B≤C<360×12×1010.

#### **Test Set 1 (Visible Verdict)**

There is a time tt that corresponds to the input such that:

* tt is an integer number of seconds after midnight.
* tt can be read from the input clock without rotating it.

#### **Test Set 2 (Visible Verdict)**

There is a time that corresponds to the input and is an integer number of seconds after midnight.

#### **Test Set 3 (Visible Verdict)**

There is a time that corresponds to the input and is an integer number of nanoseconds after midnight.

### Sample

*Note: there are additional samples that are not run on submissions down below.*

Sample Input

[save\_alt](https://codejam.googleapis.com/dashboard/get_file/AQj_6U34th0pzGfiad6_rgnKtaeO276-XKf0-_xK1f9HA4bib1OXwZKk49v7p9MfFZ0cpge-8KjlSK7nSlMjptNCnag/broken_clock_sample_ts1_input.txt?dl=1)

content\_copy

3

0 0 0

0 21600000000000 23400000000000

1476000000000 2160000000000 3723000000000

Sample Output

[save\_alt](https://codejam.googleapis.com/dashboard/get_file/AQj_6U15Vr6VK-tqD2GedQRvZbCLOMhT1_i8lgP7fCaZNwfe9x6tIBf_zWKPtm84f7B9qDEt2h6C4VWt7_lG6Qd9kXYZ/broken_clock_sample_ts1_output.txt?dl=1)

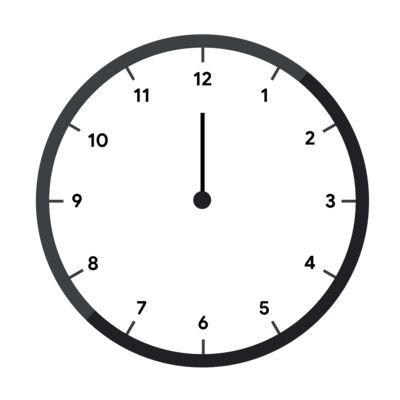
content\_copy

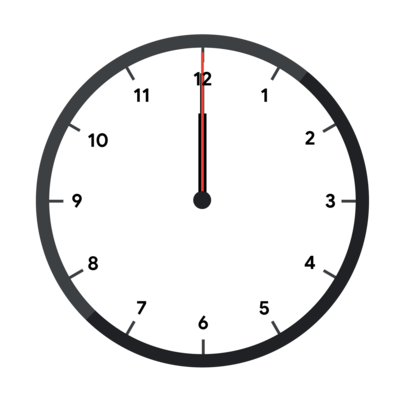
Case #1: 0 0 0 0

Case #2: 6 30 0 0

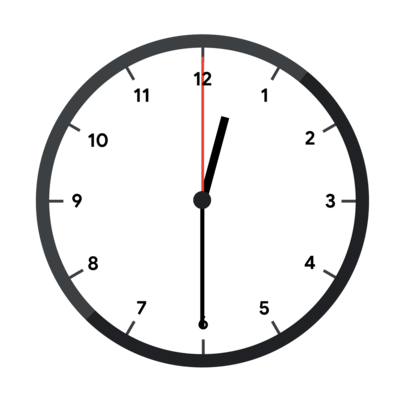
Case #3: 1 2 3 0

In Sample Case #1, all hands point up (as in the first picture below) which happens only exactly at midnight (as in the second picture below).





Sample Case #2 is the one pictured in the main part of the statement. The angles of the hands in degrees are 00, 180180 and 195195. These angles can correspond to 66⁠h 3030⁠m 00⁠s without rotating the clock, as the pictures in the main part of the statement show. Notice however, that at 00⁠h 3030⁠m 00⁠s (pictured below), the clock looks the same but rotated 180 degrees.



Even in Test Set 1, 00⁠h 3030⁠m 00⁠s would be a valid answer. The limit only says that there is one valid time that does not require rotating the clock, but times that work with rotation are also valid answers.

In Sample Case #3, the input represents the clock in the first picture below and the given output happens when interpreting the clock as in the second picture below.

### Additional Sample - Test Set 2

*The following additional sample fits the limits of Test Set 2. It will not be run against your submitted solutions.*

Sample Input

3

5400000000000 5400000000000 5400000000000

10800000000000 32400000000000 34200000000000

23076000000000 23760000000000 25323000000000

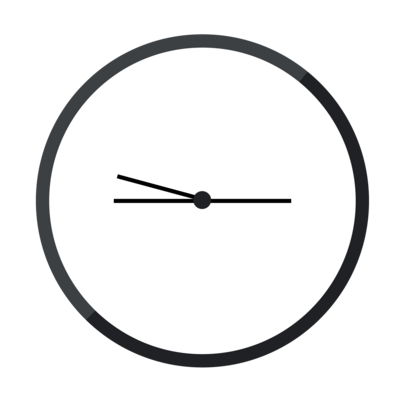
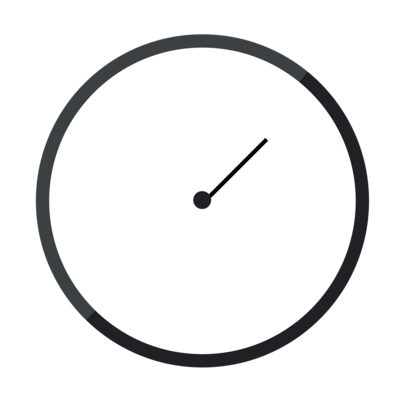
Sample Output

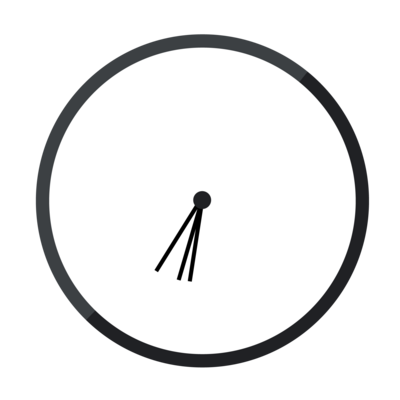
Case #1: 0 0 0 0

Case #2: 0 30 0 0

Case #3: 1 2 3 0

Sample Cases in this test set are the same as in the previous one, but the clock is rotated by 4545, 9090, and 180180 degrees clockwise respectively, as shown below.





### Additional Sample - Test Set 3

*The following additional sample fits the limits of Test Set 3. It will not be run against your submitted solutions.*

Sample Input

1

0 11 719

Sample Output

Case #1: 0 0 0 1

As explained above, 11 nanosecond after midnight the hands are moved by 11, 1212, and 720720 ticks, respectively. If the clock is also rotated counter-clockwise by 11 tick, the hand angles are exactly the ones given in the input.

**Code:**

|  |
| --- |
| **#include <bits/stdc++.h>**  **using namespace std;**  **typedef long long ll;**  **typedef long double ld;**  **#ifdef DEBUG**  **#define int128 long long**  **#else**  **#define int128 \_\_int128**  **#endif // DEBUG**  **const int128 bck11 = 15709090909091; // (4\*mod + 1) // 11**  **const int128 mod = 12\*60\*60\* (int128)1e9; // find ticks in 12 h**  **bool f(int TN, int128 a, int128 b, int128 c) {**  **int128 t = (((b-a) \* bck11) % mod + mod) % mod;**  **if ((708 \* t - c + b) % mod == 0) {**  **ll n = t % (ll)1e9;**  **t /= 1e9;**  **ll s = t % 60;**  **t /= 60;**  **ll m = t % 60;**  **ll h = t / 60;**  **cout << "Case #" << TN << ": " << h << " " << m << " " << s << " " << n << "\n";**    **return true;**  **}**  **else return false;**  **}**  **signed main() {**    **int T;**  **cin >> T;**  **for (int TN = 1; TN <= T; TN++) {**  **ll ta, tb, tc;**  **cin >> ta >> tb >> tc;**  **int128 a = ta, b = tb, c = tc;**  **if (f(TN, a, b, c) || f(TN, a, c, b) || f(TN, b, a, c) || f(TN, b, c, a) || f(TN, c, a, b) || f(TN, c, b, a)) {**    **}**  **else {**  **cout << "Problem" << "\n";**  **}**  **}**  **return 0;**  **}** |

**Logic of code:**

In this code, mod variable is total ticks in 12 hours and bck11 is inverse value of 11 which we got from (mod\*c + 1) / / 11. the perfect divisor will find when the value of c is 4.

Bool f (function)

This function will take 4 variables as a input and it will return true or false, where TN is the number of test cases and a, b, c are the values of hour, minute and second. Variable t will give as total ticks in the time which is given in input. We will calculate the value of hour, minute, second and nanosecond from the value of t because 1 tick is equal to 1/12×10−10 degrees. This means that the hours hand rotates exactly 1 tick each nanosecond, the minutes hand rotates exactly 12 ticks each nanosecond and the seconds hand rotate exactly 720 ticks each nanosecond.

Main program:

Firstly, it is taking input of total testcases after that it is taking value of three integers A, B, and C: the angles of each hand, relative to an arbitrary axis and given in ticks in the clockwise direction.

But there is a problem that hand is unknown that which hand is which. Since there are only 3 possibilities for the assignment of angles in the input to a specific hand, we can just try them all. After we have angles assigned to specific hands.

in our code this work is doing by below given code:  
(f (TN, a, b, c) || f (TN, a, c, b) || f (TN, b, a, c) || f (TN, b, c, a) || f (TN, c, a, b) || f (TN, c, b, a))  
we will try each possibilities and whichever possibility will return true (as the work of function f is returning true or false) we will print that possibility else we will print “Problem”.

**Thank You**