Problem L. Daylight Saving Time

Time limit: 1 second

Last month, xiaodao together with her friend poteko took a flight from San Francisco to Shanghai. When they were driving to the airport, xiaodao suddenly realized that the clock time on potekos car is one hour faster than the clock time on her mobile phone. And both of them might be correct, but how can it be? Because that day was Nov 6^{th}, the Daylight saving time switch day this year.

Daylight saving time (DST) or summer time is the practice of advancing clocks during summer months by one hour so that evening daylight lasts an hour longer. It is arguable that using DST can reduce overall energy consumption. Not all of us are using DST now, and for those regions adopting DST, the practices are also different.

In the case of California, effective in the U.S. in 2007 as a result of the Energy Policy Act of 2005, the local time changes from Pacific Standard Time (PST) to Pacific Daylight Time (PDT) at `02:00` to `03:00` on the **second Sunday in March** and the local time changes back from PDT to PST at `02:00` to `01:00` on the **first Sunday in November**.

Because it is one hour longer on that day, so xiaodao and poteko didn't miss the flights, but they found it still a little confusing. Interestingly, once xiaodao went back to Shanghai, she met a bug caused by exactly the same issue during the work. You might also be in trouble with DST some day, so here comes this problem and hope it will be helpful.

The local time in California without specifying whether it is PST or PDT could be ambiguous in some cases (e.g. `2016-11-06 01:25:00`). In this problem, you are given a local time in California. Check whether it is "`PST`", "`PDT`", "`Both`" or "`Neither`".

Input

The first line of the input gives the number of test cases, T.

T test cases follow. Each test case consists of one line, a date string written in the following format: `YYYY-MM-DD HH:MM:SS`.

Output

For each test case, first output one line containing "`Case #x: `", where `x` is the test case number (starting from `1`), following a result string which in one of "`PST`", "`PDT`", "`Both`" or "`Neither`".

Limits

- $1 \leq T \leq 1000$.
- date will be legal and between "`2007-01-01 00:00:00`" and "`2100-12-31 23:59:59`".

Sample input and output

Sample Input	Sample Output
4 2016-03-13 01:59:59 2016-03-13 02:00:00 2016-11-06 00:59:59 2016-11-06 01:00:00	Case #1: PST Case #2: Neither Case #3: PDT Case #4: Both