Problem 212: Time Troubles

Difficulty: Medium

Author: Brett Reynolds, Annapolis Junction, Maryland, United States

Originally Published: Code Quest 2023

Problem Background

Time may not actually be money, but it's still very important when it comes to organizing our daily lives. Unfortunately, when working with other people around the world, times tend to be different. What's early in the morning in New Zealand is late at night in France. The world is divided into dozens of different time zones, and being able to work between these time zones is often a problem in computer programming.

Problem Description

Your team is working on a scheduling application that will be used at multiple Lockheed Martin sites around the world. Users of the application will want to see events displayed in their own time zone, but in order to keep everything organized, your application will store times of events in the UTC time zone (Coordinated Universal Time… the acronym is based on French, if you're wondering).

Each time zone in the world is defined using a UTC offset - the number of hours by which that time zone is ahead or behind of UTC. For example, most of the Eastern coast of the United States has a UTC offset of -4 hours for much of the year (written as UTC-4). This means that when it's 11:00 AM in the UTC time zone, it's 7:00 AM in the US Eastern time zone (11 - 4 = 7).

Your program will be given a time entered by the user, and the UTC offset of the user's time zone. You'll need to convert that time from the user's time zone into UTC for storage in your system's database.

Sample Input

The first line of your program's input, **received from the standard input channel**, will contain a positive integer representing the number of test cases. Each test case will include the following values, separated by spaces:

- The date of the event, formatted as MM/DD/YYYY, where "MM" is the two-digit month (January is 01, December is 12), "DD" is the two-digit day of the month, and "YYYY" is the four-digit year
- The time of the event, formatted as HH:MM, using the 24-hour clock and two digits for both numbers (00:00 is 12:00 AM, 23:59 is 11:59 PM)
- A number representing the offset of the user's time zone from UTC, in hours. This number may contain decimal values.

```
3
08/17/2019 12:00 8
08/17/2019 12:00 -4
08/01/2019 04:15 7.5
```

Sample Output

For each test case, your program must output the date and time of the event, converted to the UTC time zone using the given hour offset. The date and time must be formatted as “MM/DD/YYYY HH:MM” as shown in the input.

```
08/17/2019 04:00
08/17/2019 16:00
07/31/2019 20:45
```