Littlepay Coding Exercise

Welcome to the Littlepay coding exercise!

Please complete the exercise below and send us your solution. We ask you to complete this work so that we get a feel for how you approach a problem, how you think, and how you design and test your code. You are welcome to use any libraries you see fit to use.

We look forward to seeing what you can do!

The Littlepay Engineering team

Littlepay is a cloud service that enables transit companies to charge their passengers for their travel by credit card. In order to do this at scale, we use a pretty sophisticated tech stack and have to solve some pretty interesting problems. To keep this exercise small so we don't take too much of your time, you will be building a hypothetical system that uses some highly simplified concepts from our system.

How the hypothetical system works

Before boarding a bus at a bus stop, passengers tap their credit card (identified by the PAN, or Primary Account Number) on a card reader. This is called a tap on. When the passenger gets off the bus, they tap their card again. This is called a tap off. The amount to charge the passenger for the trip is determined by the stops where the passenger tapped on and tapped off. The amount the passenger will be charged for the trip will be determined as follows:

Trips between Stop 1 and Stop 2 cost $3.25
Trips between Stop 2 and Stop 3 cost $5.50
Trips between Stop 1 and Stop 3 cost $7.30

Note that the above prices apply for travel in either direction (e.g. a passenger can tap on at stop 1 and tap off at stop 2, or they can tap on at stop 2 and can tap off at stop 1. In either case, they would be charged $3.25).

Completed Trips
If a passenger taps on at one stop and taps off at another stop, this is called a complete trip. The amount the passenger will be charged for the trip will be determined based on the table above. For example, if a passenger taps on at stop 3 and taps off at stop 1, they will be charged $7.30.

Incomplete trips
If a passenger taps on at one stop but forgets to tap off at another stop, this is called an incomplete trip. The passenger will be charged the maximum amount for a trip from that stop to any other stop they could have travelled to. For example, if a passenger taps on at Stop 2, but does not tap off, they could potentially have travelled to either stop 1 ($3.25) or stop 3 ($5.50), so they will be charged the higher value of $5.50.

Cancelled trips
If a passenger taps on and then taps off again at the same bus stop, this is called a cancelled trip. The passenger will not be charged for the trip.

The problem

Given an input in JSON format containing an array of 'taps' where each tap represents a single tap on or tap off you will need to create an output containing trips made by customers.

taps.json [input]
You are welcome to assume that the input is well formed and is not missing data.

Example input:

```
{
   "taps": [
      {
         "id": 1,
         "datetimeUTC": "22-01-2021 13:00:00",
         "tapType": "ON",
         "stopId": "Stop1",
         "companyId": "Company1",
         "busId": "Bus37",
         "primaryAccountNumber": "5500005555555559"
      },
      {
         "id": 2,
         "datetimeUTC": "22-01-2021 13:05:00",
         "tapType": "OFF",
         "stopId": "Stop2",
         "companyId": "Company1",
         "busId": "Bus37",
         "primaryAccountNumber": "5500005555555559"
      },
      ...
   ]
}
```

trips.json [output]
You will need to match the tap ons and tap offs to create trips. You will need to determine how much to charge for the trip based on whether it was complete, incomplete or cancelled and where the tap on and tap off occurred. You will need to write the trips out in JSON format.

Example output:

```
{
   "trips": [
      {
         "started": "22-01-2021 13:00:00",
         "finished": "22-01-2021 13:05:00",
         "durationSecs": 900,
         "fromStopId": "Stop1",
         "toStopId": "Stop2",
         "chargeAmount": 3.25,
         "companyId": "Company1",
         "busId": "B37",
         "primaryAccountNumber": "5500005555555559",
         "status": "COMPLETED"
      },
      ...
   ]
}
```

Important: Do not use real credit card numbers (PANs) in the test data you provide to us.

You can find credit card numbers suitable for testing purposes at
http://support.worldpay.com/support/kb/bg/testandgolive/tgl5103.html

As part of your solution:

- List any assumptions that you made in order to solve this problem
- Provide instructions on how to run the application
- Provide a test harness to validate your solution

Thank you and have fun!