stripe HackerRank Assessment 2018 21m to test end

1

2

☆ Platform balance

Stripe Connect allows marketplaces and platforms, like the KickStarters, Shopifys, and Postmates of the world, to accept money and pay out to third parties. Let's say TaxiCo, an on demand ride-sharing service, wants to pay out its drivers for each ride they complete. They could simply use Stripe's Connect API to make that happen. Each driver could have their own Stripe account, and TaxiCo could then send the funds to that driver's account (less the Stripe fee and the cut TaxiCo takes).

Stripe's standard fees in the US are 2.9% + $0.30 on each transaction. If a ride costs $10, Stripe takes $0.59 (`$0.59 = $10 * 2.9% + $0.30`). TaxiCo would be left with $9.41 to split between itself and the driver. TaxiCo can then specify to Stripe how much to transfer to the driver. For example, TaxiCo could tell Stripe to send the driver $8.77. Once the driver has been paid, Stripe transfers the remaining $0.64 into TaxiCo's Stripe account.

For this problem, we'd like you to build a system that tracks the balances for TaxiCo and its drivers. We're going to communicate on `stdin` and `stdout` rather than using HTTP, to simplify things.

Your program should:

1. Take an array of lines as input and return an integer array equal in size to the number of lines that begin with 'BAL:' (see examples below).
2. If an input line begins with `API:`, update the balance for the platform (in this case, TaxiCo), and if present, the destination account (i.e. the driver's account). TaxiCo and each driver account will have a unique merchant ID.
3. If an input line begins with `BAL:`, append the current balance in USD cents for the account in question to the output array.
4. Assume that we are using the U.S. standard Stripe fee of 2.9% + $0.30.

Here's the previous case as an example:

Example of TaxiCo payout to the driver's account (i.e. the destination account)

Input:

```
3

API:
```

```
BAL: merchant=111111
```

Output:

```
64
877
```

Some more details:

- Amounts are in USD cents as integers (e.g. amount=1000 => $10.00).
- The first line of the sample input will be the number of lines that follow. Your sample code should already deal with this for you.
- Lines on `stdin` will always begin with `API:` or `BAL:` followed by a space and a URL-encoded string.
- We will not test your solution with malformed input, and you may handle it however you see fit.
- The `merchant` field of `BAL:` requests can correspond to either a driver or TaxiCo's account ID. These account IDs are globally unique.
- You should preserve the original ordering of the `BAL:` requests, when outputting balances.
- Track balance amounts to the nearest hundredth of a cent throughout the calculation. The final answer of a `BAL:` request should be an integer; round to the nearest cent (i.e. 0.00 -> 0.49 round down, 0.50-0.99 round up).
- The program should track the current balance for TaxiCo and its drivers. In other words, you may want to consider using a data structure to store each account and its corresponding total balance.
- If there is not a destination account, the full payment minus fees should go to the merchant.
- We recommend the following libraries for parsing url params. You may look up these libraries or use resources to find out how to parse urls in your language:
  - **Python:** `urllib.parse`
  - Ruby: `CGI`
  - Java: `org.apache.http.client.utils.URLEncodedUtils`

Example without a destination account:

Input:

```
BAL: merchant=10101010
```

1

2

Output:

```
1912
```

Example of TaxiCo collecting fees for multiple rides:

Input:

```
4

API:
amount=1000&merchant=123456789&destination[account]=111111&destination[
amount]=877

API:
amount=800&merchant=123456789&destination[account]=112211&destination[a
mount]=622

BAL: merchant=123456789

BAL: merchant=112211
```

Output:

```
189

622
```

YOUR ANSWER

We recommend you take a quick tour of our editor before you proceed. The timer will pause up to 90 seconds for the tour. Start tour ✖

Draft saved 11:13 pm Original code Java 8 stripe HackerRank Assessment 2018

21m
to test end

1

2

```java


public class Solution {

    // Complete the balance function below.
    static List<Integer> balance(List<String> lines) {


    }


    public static void main(String[] args) throws IOException {
}

```

Line: 24 Col: 1

Test against custom input

Run Code

Submit code & Continue

(You can submit any number of times)

Download sample test cases *The input/output files have Unix line endings. Do not use Notepad to edit them on windows.*

About Privacy Policy Terms of Service