YCHARTS Modern Financial Platform

Question 1:

DATA STRUCTURES

Implement a hashmap (with amortized constant time look-ups) in Python without using a hashmap primitive. Please include an executable testing framework for your data structure.

YCHARTS Modern Financial Platform

Question 2:

FRONT-END

Create a To-Do Widget.

Requirements:

- The initial state should be an empty input box, and an ADD button.
- If the user enters input text to the box, clicking ADD should add that text to a list.
- There should be a way to delete a list element.
- On form submit, the list elements should be submitted as an array.
- You should be able to have more than one component per page.
- Does not accept empty or white space input
- Does not accept illegal characters defined by (^.*?(?=[\^#%&$*:<>\?\^{\|\}]).*$)

Please use only straight HTML, CSS, and Javascript. Do not use any CSS frameworks or external Javascript libraries.

Question 3:

BACK-END

The Problem:

Reconciliation is a term YCharts uses for a set of correctness and consistency measures applied to the data we receive and use in financial calculations.One of the most common reconciliation checks is called *unit reconciliation*, which answers the question, "does the transaction history add up to the number of shares the bank says I have?". For example, if the bank said I had 100 shares of Apple at the end of yesterday, and I bought 20 shares of Apple today, then we expect the bank to report 120 shares at the end of today. This surprisingly isn't always the case! The bank may send incomplete data, we may be parsing it incorrectly, or there may be events like corporate actions or trade settlement lag that cause an inconsistency.

The Input:

recon.in has three sections:

D0-POS describes the positions in the account at the end of Day 0. Each record is a space-separated pair of Symbol and Shares. For example "AAPL 10" means 10 shares of AAPL were held at the end of Day 0, and "Cash 122.16" means we had $122.16 in the account at the end of Day 0.

D1-TRN describes the transactions that occurred in the account on Day 1. Each record is space-separated collection of four items: Symbol, Transaction Code, Shares, and Total Value. For example, the record "AAPL BUY 10 6123.21" means 10 shares of AAPL were bought for a total cost of $6123.21.

D1-POS describes the positions in the account at the end of Day 1, and has the same format as D0-POS.

The Output:

The objective is to write a program that produces recon.out. Each line should be a space-separated record indicating a position that fails unit reconciliation. For example, "AAPL 10" means that the reported shares of AAPL in D1-POS is 10 higher than expected based on the transactions.

FAQ:

1. Can you sell a stock you don't own already ?
Yes, you can. In finance this is called a short sale. The goal here is to compare data difference simply, so it is acceptable to have a negative position.

2. Do I need to validate the data format?
No, assume the data format in recon.in has no errors (WYSIWYG).

(Continue to Next Page)

Question 3 (cont):

BACK-END

Test Data:

recon.in

D0-POS
AAPL 100
GOOG 200
SP500 175.75
Cash 1000

D1-TRN
AAPL SELL 100 30000
GOOG BUY 10 10000
Cash DEPOSIT 0 1000
Cash FEE 0 50
GOOG DIVIDEND 0 50
TD BUY 100 10000

D1-POS
GOOG 220
SP500 175.75
Cash 20000
MSFT 10

recon.out

Cash 8000
GOOG 10
TD -100
MSFT 10
