ServiceNow Take Home Assignment

Welcome! Before diving into the coding problem, we want to tell you our goals & expectations for this assignment.

Goals

These are our goals with the assignment:

- Allow you to work on your own without typical in-person interview pressures
- Give you freedom in how you approach the problem & solution
- Give us a view on how you'd solve a problem that, although greatly simplified, matches the spirit of problems we solve on our team

Evaluation Criteria

We will evaluate your solution by answering three questions:

1. **Does the solution run successfully on our machines?**

   Your solution should be available via a downloadable .zip file or GitHub/BitBucket repo along with instructions for how to run it. For whatever runtime/dependencies you pick, specify the version so we don't hit compatibility problems.

2. **Does the solution correctly solve the problem?**

   Your submission should correctly solve the problem according to the problem statement & constraints.

3. **Is the code production-ready?**

   Production-ready means different things to different people and different projects. For this assignment, production-ready means that the code meets your standards for production quality code, which may include:
   - Good names & minimal duplication
   - Cohesive functions & modules
   - Adequate automated test coverage

Now that goals and expectations are in place, let's get to the problem!

The Problem: Software Spend Reporter

Your assignment is to build the first iteration of a product that helps IT departments make sense of their software spend. It will be a console application that reads in software spend data and outputs a report summarizing that data.

The input file is a comma separated values (CSV) file that contains a list of software spend transactions. It has four columns:

- **Transaction Date:** When the purchase was made
- **Vendor:** The vendor that was paid in this transaction
- **Product:** The software product that was purchased
- **Amount:** The amount, in US Dollars, that was spent in this transaction

The expected console output is a two-level tree that aggregates the spend for vendors and their products. To best illustrate this output, here is an example:

Example Input file:

Transaction Date	Vendor	Product	Amount
1/28/2019	Microsoft	Office365	432854
1/27/2019	Adobe	Creative Cloud	98445
1/24/2019	Amazon	AWS	12443
1/19/2019	Microsoft	Azure	5332
1/11/2019	Adobe	Illustrator	1233
12/24/2018	Amazon	AWS	11977
12/5/2018	Box	Box	66122
11/24/2018	Amazon	AWS	9322
11/3/2018	DocuSign	DocuSign	45221
1/28/2018	Microsoft	Office365	389777

Example usage (Bash example):

```
>> ./software_spend_reporter /path/to/inputfile.csv

Adobe $99,678
  Creative Cloud $98,445
  Illustrator $1,233
Amazon $33,742
  AWS $33,742
Box $66,122
  Box $66,122
DocuSign $45,221
  DocuSign $45,221
Microsoft $827,963
  Azure $5,332
  Office365 $822,631
```

Here are more specific requirements for the console application:

- Like the example above, we should be able to run the executable with an input parameter that specifies the path to the input csv file. You can choose the executable name and Operating System to run it on.
- The vendor nodes in the tree:
  - have no indentation
  - contain the total spend one space after the vendor name formatted properly in US dollar and number formats.
  - are sorted alphabetically
- The product nodes in the tree:
  - are indented two spaces under their corresponding vendor node
  - are sorted alphabetically within their vendor node
  - contain the total spend one space after the product name. This spend is formatted properly in US dollar and number formats.
- Don’t worry about properly handling all error cases. You can assume the user will run the tool against a valid input file. We’d definitely handle these cases in a real application, but we don’t want to have this assignment take too much of your time.
- Use whatever language, framework, libraries, packages, and SDKs you feel most comfortable with that help you solve the problem. Just make sure we know how to build and run your app. Our ability to build and run without needing additional support is part of our evaluation of your submission, so make sure it works!
- Good luck and have fun!