Assignment 4

You will be writing an inventory system that is backed by JSON data and will be working with a starter file that contains a JSON string. The code you write will need to follow the following guidelines.

The what

You're at work one day and your boss asks you about that fancy programming language you've been learning, Python. She asks you if you can use it to read JSON data from a supplier and build an inventory. "Of course!" you say. She tells you that your parts supplier, Midwest Widget Co, has sent the company JSON data that needs to be loaded and then checked against orders. You'll need to load the JSON data, and then take input from a user to order new parts. If the part is not in stock, or the number of parts is greater than the available inventory, you need to alert the user and show them that its either out of stock or that there are less available than requested. Once the user is done entering in their order, you'll need to package up their order in JSON format, since that's what Midwestern Widget Co uses after all!

Specifics

The JSON data is comprised of a few nested dictionaries. In the top level, there is a key called "parts" that maps to a list that contains all the parts available for ordering. Each of those parts is also a key into the dictionary. That key maps to dictionary that has two values, quantity and price. The quantity maps to an integer with the number of parts available for order, the price maps to a float with the price of the part. A sample JSON object is provided below:

```
{
   "parts":[
      "widget",
      "sprocket",
      "thing-a-ma-bob"
   ],
   "widget":{
      "price":0.99,
      "quantity":74
```

```
    },
    "sprocket":{
        "price":3.99,
        "quantity":123
    },
    "thing-a-ma-bob":{
        "price":1.78,
        "quantity":57
    }
}
```

The strings in the list in parts are the keys for the rest of the data in the dictionary. Those keys map to separate dictionaries with the prices and quantities. The json string is provided in the code, it is called supplier_data.

The ordering system will work like this:

Prompt the user for input, and indicate they can enter in the word “quit” to quit. The user should enter in a part and then the quantity on two separate lines (so you’ll need two input statements). After both pieces of information have been entered in, check if the order is allowed or not, and display an error message with the appropriate information (part doesn’t exist, part exists but not enough quantity). If the order is valid, store it and continue. Once the user enters quit, print out an order summary showing the part, number ordered, the price per part and total per part with a grand total at the end. The order data MUST be stored in a dictionary. You must also allow the user to order a part more than once and validate that both orders are not exceeding the total amount available!

Sample Output

Your program should produce output that is very similar if not identical to this. Major deviations from the formatting will result in points lost. The error handling behavior should be the same.

Welcome to the parts ordering system, please enter in a part name, followed by a quantity

Parts for order are:

sprocket

gizmo

widget

dodad

Please enter in a part name, or quit to exit: blargh

Error, part does not exist, try again

Please enter in a part name, or quit to exit: quit

Your order

Total: $0

Thank you for using the parts ordering system!

And another run with an actual order

Welcome to the parts ordering system, please enter in a part name, followed by a quantity

Parts for order are:

sprocket

gizmo

widget

dodad

Please enter in a part name, or quit to exit: gizmo

Please enter in a quantity to order: 1000

Error, only 2 of gizmo are available!

Please enter in a part name, or quit to exit: gizmo

Please enter in a quantity to order: 1

Please enter in a part name, or quit to exit: sprocket

Please enter in a quantity to order: 15

Please enter in a part name, or quit to exit: widget

Please enter in a quantity to order: 1

Please enter in a part name, or quit to exit: quit

Your order

gizmo - 1 @ 7.98 = 7.98

sprocket - 15 @ 3.99 = 59.85

widget - 1 @ 14.32 = 14.32

Total: $82.15

Thank you for using the parts ordering system!

Another run with multiple orders for a single part

Welcome to the parts ordering system, please enter in a part name, followed by a quantity

Parts for order are:

sprocket

gizmo

widget

dodad

Please enter in a part name, or quit to exit: gizmo

Please enter in a quantity to order: 2

Please enter in a part name, or quit to exit: gizmo

Please enter in a quantity to order: 2

Error, only 0 of gizmo are available!

Please enter in a part name, or quit to exit: quit

Your order

gizmo - 2 @ 7.98 = 15.96

Total: $15.96

Thank you for using the parts ordering system!

Tips and Hints

1. Start on something small and get it working before moving on. Don't try and tackle it all at once!
2. You'll need to validate the input between inputs
3. You can use print() with no arguments to print out extra blank lines
4. A while loop is ideal for the main program loop
5. You'll need to check if a part has already been ordered before checking if sufficient quantity is available. If it has been ordered, you'll need to account for that when checking the inventory levels
6. supplier_data in the provided code is a JSON string. You need to convert it to a dict with json.loads

Good luck!