Name: Ajay

Problem: I figured out that the best way to store the data would be in a dictionary where the access to the different purchases that one made would be based on a key value. I think that the best way to do this key would be the date that the item was purchased.

File with solution:
possible_user_input.py

Name: Ellis

Problem: The problem that I set out to solve was trying to create a time series chart based on a CSV file that we are turning into a dataframe. Along with that, creating a new dataset for the average amount that an individual spends on a specific item.

Main Points: I figured that the best way to handle this problem is by keeping the data the same and not using any aggregate functions. In the chart, the date can serve as the x-axis while the price takes on the y-axis. However, the biggest issue that I am facing is that I don't really know how to plot data using python.

Name: Dennis
Problem: I was thinking of creating a method that would predict the next item that the person will buy based off frequencies and or time of date.

Main points: The idea here is we would add up all the occurrences of the particular expense being bought and then find the probability of the person buying it. Basically it will be the frequency of the expense/ all expenses and then using a random generator we create the next potential. Another thing I wanted to try and implement is to create an algorithm that can find patterns based off the person paying expenses on a specific day a few days in a row.

Name: Zain

Problem: The problem that I chose to solve is a rather interesting one. I want the budgeting program to be able to tell if buying another item is a viable option with the average budget found in the last week.

Main points: predicting this would be very difficult, but if there is a max amount that the user cannot go over, we can look at the previous week's average budget, and determine if the new item that the user would like to addon is a viable option. If it does not go over the max amount, then the program will allow it, but say that it is a less-than-optimal decision. If this new item added on equals less than the previous week's average amount spent, then the program will say that it is a good idea. Finally, if the new item added onto the budget is more than the average budget, then the program will let the user know that this addition is a bad idea, and would hurt the budget.