Ονοματεπώνυμο: Παντελεήμων Μαλέκας
A.M: 1115201600268

Comments regarding assignment 0

1. addition.py

   A simple function that returns the sum of arguments a and b.
   It doesn't have a main method since it's meant to be run by autograder.py

2. buyLotsOfFruit

This function takes an orderList as an argument, meaning a list with tuples of two items: fruit type and its cost. It checks if a fruit is not in the fruitPrices dictionary in order to quit its execution. Otherwise it calculates the cost of each fruit by multiplying its price from the orderList and its value from the fruitPrices dictionary. It then adds each cost to the total amount that will be returned.

Also, the desired message of when a fruit is not in fruitPrices simply appears before the function is called, because a check is made in the main method. Either way, the function will exit normally if the fruit is not in fruitPrices.

3. shopSmart.py

This function calculates the cost of the orderList from each shop (through the use of the getPriceOfOrder function). Then it proceeds to append the cost in a ListofCosts list which contains every shop name along with its price of the order. After that it finds the cheapest cost from the prices we have and it returns the shop name that contains this price. If something went wrong and we didn't find any cheap cost the function returns None.

4. Stack.py

In this program a Stack class is defined along with its basic functions: Push, Pop, Empty. It will be used to find out if a string with brackets (i.e.: [(){}]) is of equal weight.

We check if our string contains only the characters we want, if it doesn't the program finishes the execution. Also if our string begins with a closing bracket, we can't have equal weight so the program exits there too.

If the programs sees an opening bracket it pushes it in our Stack. If it sees a closing bracket it pops the item at the top, if the Stack is not empty. It also compares the item it popped with the closing bracket to see if they are of the same type, i.e. [with]. If it's not, we can't have equal weight so the program exits.. Also if our Stack is empty and we have a closing bracket this means our string is not of equal weight. Finally, after iterating the string we check whether our Stack is empty or not. If it is, our string is of equal weight. Otherwise this means that at least one character didn't match with anything, so our string is not of equal weight.