Lab *0*

Tasnimul Hasnat

190041113

Monday, Aug 21 2023

Addition

```
def add(a, b):
    "Return the sum of a and b"
    "*** YOUR CODE HERE ***"

    return a+b
```

Basic addition of two numbers, nothing complicated.

buyLotsOfFruit function

```
def buyLotsOfFruit(orderList):
    """
    orderList: List of (fruit, numPounds) tuples

    Returns cost of order
    """
    totalCost = 0.0
    "*** YOUR CODE HERE ***"
    for fruit, numPounds in orderList:
        if fruit not in fruitPrices:
            return None
        totalCost += fruitPrices[fruit] * numPounds

    return totalCost
```

The task is to implement a function that takes an order list of (fruits, numPounds) tuples and returns the total cost of the items in the order list. The solution is to iterate through the list and compute the product of the number of pounds of each fruit ordered with its per pound price present in the predefined dictionary. The sum total of these products are stored in the *total*Cost variable and returned at the end of the function. Also there is a check for any fruits not in the list.

shopSmart Function

```
def shopSmart(orderList, fruitShops):
    """
    orderList: List of (fruit, numPound) tuples
    fruitShops: List of FruitShops
    """
    "*** YOUR CODE HERE ***"

    bestPrice = float('inf')
    bestShop = None
    for shop in fruitShops:
        price = shop.getPriceOfOrder(orderList)
        if price < bestPrice:
            bestPrice = price
            bestShop = shop
    return bestShop
```

The task requires us to find the shop for which the cost of the order is the least. The *shopSmart* function has two parameters – *orderList*, a list of tuples of two elements, and *fruitShops*, a list of fruit shops. The function iterates through each fruit shop, computes the price of the given fruit order using a predefined method called *getPriceOfOrder*, and then compares these prices to determine the shop offering the lowest price. In the end, it returns the *bestShop*.

—xxx—