AU 332 Artificial Intelligence: Principles and Techniques

By: Chi Zhang (517021910658)

HW#: 0

September 15, 2019

I. INTRODUCTION

A. Purpose

As required by Question 1, 2 and 3, my function need to implement following features :

- Question 1: Enter a and b, and return the sum of a and b
- Question 2: Enter a list of `(fruit, pound)` tuples and return the cost of the fruit in the list. If the list contains some fruit that not in the `fruitPrices`, the funtion will print an error message and return **None**
- Question 3: Enter a list of `(fruit, pound)` tuples and a list of `fruitshop`, and return the shop which the account is least.

The homework aims to help us familiar with Python, especially the usage of function and class.

B. Equipment

There is a minimal amount of equipment to be used in this lab. The few requirements are listed below:

- Python 3.7.0 (Anaconda)

C. Procedure

1. Problem 1

The function of Problem 1 is simple, I just return `(a+b)` .

```
def add(a, b):
    ”print the sum of a and b”
    ”*** YOUR CODE HERE ***”
    return (a+b)
```

2. Problem 2

The function **buyLotsOfFruit** receive a list of (fruit, numPounds) tuples, which means we can use the *for* statement to read all the fruit and weight in the list

In my code, I use `fruit` to represent the element in the list, therefore, fruit[0] is the variety of fruit and fruit[0] is the weight of this fruit. Before I compute the cost of each kind of fruit, I use the *if* statement to judge if this fruit is in the `fruitPrices`. If not, the function will print an error message and the function will return **None** as result.

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

    Returns cost of order
    ”””
    totalCost = 0.0
    ”*** YOUR CODE HERE ***”
    for fruit in orderList:
        if fruit[0] in fruitPrices:
            totalCost += fruitPrices[fruit[0]] * fruit[1]
        else :
            totalCost = None
            print(’error! ’+fruit[0]+ ’ is not in the fruitPrices list’)
            break

    return totalCost
```

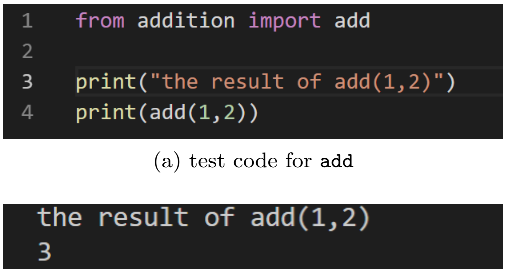
3. Problem 3

The file `shop.py` has defined class `FruitShop`, which includes function `getCostPerPound` and `getPriceOfOrder`. In my code, I use `the_shop` to represent the shop in the `fruitShops` list. In each loop, `the_shop.getPriceOfOrder` will compute the cost of the order in current shop. Besides, I create variable `least_amount` and `correspond_shop` to record the lowest cost and corresponding name of shop.

```
def shopSmart(orderList, fruitShops):
    ”””
        orderList: List of (fruit, numPound) tuples
        fruitShops: List of FruitShops
    ”””
    ”*** YOUR CODE HERE ***”
    least_amount = None
    correspond_shop = None
    for the_shop in fruitShops:
        current_amount = the_shop.getPriceOfOrder(orderList)
        if least_amount is None or least_amount > current_amount :
            least_amount = current_amount
            correspond_shop = the_shop

    return correspond_shop
```

II. EXPERIMENT

This section consists of screenshots taken during the laboratory procedure.



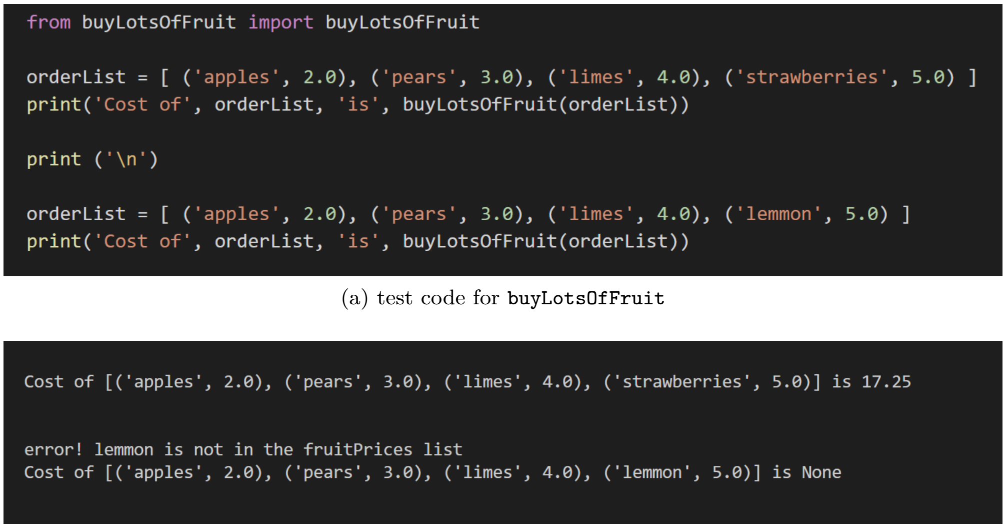
```python
from addition import add

print("the result of add(1,2)")
print(add(1,2))
```

(a) test code for `add`

```
the result of add(1,2)
3
```

(b) result of the test

FIG. 1: Figures of the test for question 1



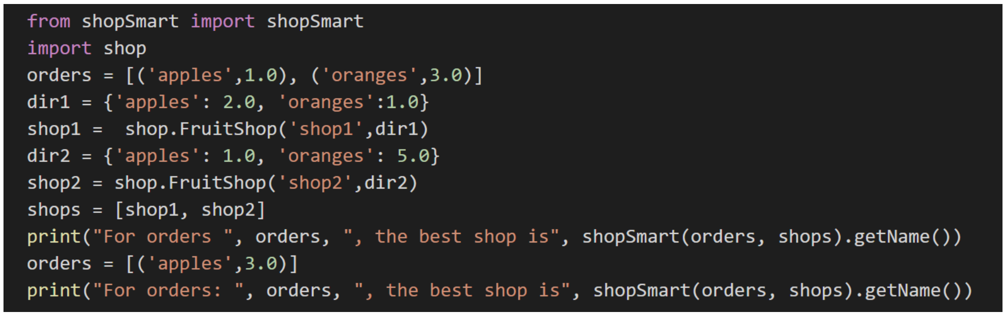
```python
from buyLotsOfFruit import buyLotsOfFruit

orderList = [ ('apples', 2.0), ('pears', 3.0), ('limes', 4.0), ('strawberries', 5.0) ]
print('Cost of', orderList, 'is', buyLotsOfFruit(orderList))

print ('\n')

orderList = [ ('apples', 2.0), ('pears', 3.0), ('limes', 4.0), ('lemmon', 5.0) ]
print('Cost of', orderList, 'is', buyLotsOfFruit(orderList))
```

(a) test code for `buyLotsOfFruit`

```
Cost of [('apples', 2.0), ('pears', 3.0), ('limes', 4.0), ('strawberries', 5.0)] is 17.25


error! lemmon is not in the fruitPrices list
Cost of [('apples', 2.0), ('pears', 3.0), ('limes', 4.0), ('lemmon', 5.0)] is None
```

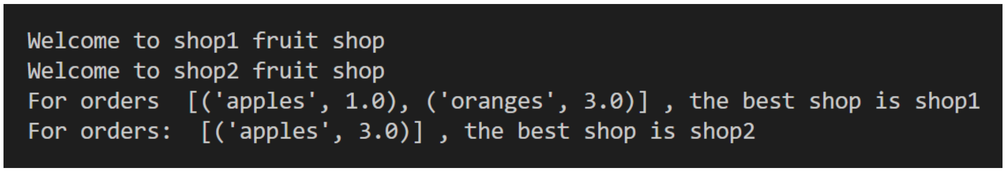
(b) result of the test

FIG. 2: Figures of the test for question 2



```
from shopSmart import shopSmart
import shop
orders = [('apples',1.0), ('oranges',3.0)]
dir1 = {'apples': 2.0, 'oranges':1.0}
shop1 =  shop.FruitShop('shop1',dir1)
dir2 = {'apples': 1.0, 'oranges': 5.0}
shop2 = shop.FruitShop('shop2',dir2)
shops = [shop1, shop2]
print("For orders ", orders, ", the best shop is", shopSmart(orders, shops).getName())
orders = [('apples',3.0)]
print("For orders: ", orders, ", the best shop is", shopSmart(orders, shops).getName())
```

(a) test code for `shopSmart`



```
Welcome to shop1 fruit shop
Welcome to shop2 fruit shop
For orders  [('apples', 1.0), ('oranges', 3.0)] , the best shop is shop1
For orders:  [('apples', 3.0)] , the best shop is shop2
```

(b) result of the test

FIG. 3: Figures of the test for question 3

III. DISCUSSION & CONCLUSION

The homework is not difficult. It is aimed to help students get accustomed to *Python*, and learn how to write reports with *Latex*. I am unfamiliar with the usage of *class* in *Python*, and I know little about *Latex*, so I spent lots of time on this homework.