Fall 2025 CS5368 Intelligent Systems Project 0 Report

First Name	Scott
Last Name	Weeden
Student ID	R12041454

Please answer the following questions and submit them through Blackboard. Be sure to submit it to Project 0 report submission. DO NOT write the report by hand and submit a scanned version. Just write the answers in a Word document and submit it. Both word and PDF submissions are accepted.

1. What did Problem 1 ask for? How did you implement it?

   Problem 1 asked to implement the add function in addition.py that takes two parameters a and b and returns their summation. The implementation was straightforward - I implemented it by simply returning a + b. The function signature was already provided, and I just needed to replace the placeholder code with the actual addition operation.

2. What did Problem 2 ask for? How did you implement it?

   Problem 2 asked to implement a buyLotsOfFruit(orderList) function in buyLotsOfFruit.py that takes a list of (fruit, pound) tuples and returns the total cost of the order. If any fruit in the list doesn't appear in the fruitPrices dictionary, it should print an error message and return 0.0 for the entire order. I implemented this by:
   - Iterating through each (fruit, numPounds) tuple in the orderList
   - Using fruitPrices.get(fruit) to safely check if the fruit exists in the price dictionary
   - If a fruit is not found (returns None), printing an error message and returning 0.0
   *- Otherwise, calculating the cost as numPounds * price_per_pound and adding it to the total*
   -Returning the total cost

3. What did Problem 3 ask for? How did you implement it?

   Problem 3 asked to implement the shopSmart(orders, shops) function in shopSmart.py that takes an order List and a list of FruitShop objects, then returns the FruitShop where the order costs the least amount in total. I implemented this by:
   - Initializing variables to track the best shop and lowest cost

- Iterating through each fruit shop in the fruitShops list
-Getting the cost of the order from each shop using fruit_shop.getPriceOfOrder(orderList)
- Comparing costs and updating the best shop whenever a lower cost is found
-Returning the shop with the lowest total cost

4. What happens when you run the following code?

python autograder.py -q q3

```
(base) owner@lorentz CS5368_Project0 % python autograder.py -q q3

/Users/owner/Downloads/CS5368_Project0/autograder.py:280: SyntaxWarning: invalid escape sequence '\.'
  tests = filter(lambda t: re.match('[^#~.].*\.test\Z', t), os.listdir(subdir_path))
/Users/owner/Downloads/CS5368_Project0/autograder.py:281: SyntaxWarning: invalid escape sequence '\.'
  tests = map(lambda t: re.match('(.*)\.test\Z', t).group(1), tests)
/Users/owner/Downloads/CS5368_Project0/autograder.py:383: SyntaxWarning: invalid escape sequence '\.'
  moduleName = re.match('.*?([^/]*)\.py', cp).group(1)
/Users/owner/Downloads/CS5368_Project0/autograder.py:385: SyntaxWarning: invalid escape sequence '\.'
  moduleName = re.match('.*?([^/]*)\.py', options.testCaseCode).group(1)
Starting on 9-12 at 16:09:44

Question q3
===========

Welcome to shop1 fruit shop
Welcome to shop2 fruit shop
*** PASS: test_cases/q3/select_shop1.test
***     shopSmart(order, shops) selects the cheapest shop
Welcome to shop1 fruit shop
Welcome to shop2 fruit shop
*** PASS: test_cases/q3/select_shop2.test
***     shopSmart(order, shops) selects the cheapest shop
Welcome to shop1 fruit shop
Welcome to shop2 fruit shop
Welcome to shop3 fruit shop
*** PASS: test_cases/q3/select_shop3.test
***     shopSmart(order, shops) selects the cheapest shop

### Question q3: 10/10 ###


Finished at 16:09:44

Provisional grades
==================
```

python autograder.py -t test_cases/q3/select_shop1

```
(base) owner@lorentz CS5368_Project0 % python autograder.py -q q3

/Users/owner/Downloads/CS5368_Project0/autograder.py:280: SyntaxWarning: invalid escape sequence '\.'
  tests = filter(lambda t: re.match('[^#~.].*\.test\Z', t), os.listdir(subdir_path))
/Users/owner/Downloads/CS5368_Project0/autograder.py:281: SyntaxWarning: invalid escape sequence '\.'
  tests = map(lambda t: re.match('(.*)\.test\Z', t).group(1), tests)
/Users/owner/Downloads/CS5368_Project0/autograder.py:383: SyntaxWarning: invalid escape sequence '\.'
  moduleName = re.match('.*?([^/]*)\.py', cp).group(1)
/Users/owner/Downloads/CS5368_Project0/autograder.py:385: SyntaxWarning: invalid escape sequence '\.'
  moduleName = re.match('.*?([^/]*)\.py', options.testCaseCode).group(1)
Starting on 9-12 at 16:09:44

Question q3
===========

Welcome to
Welcome to
*** PASS: test_cases/q3/select_shop1.test
***     shopSmart(order, shops) selects the cheapest shop
Welcome to shop1 fruit shop
Welcome to shop2 fruit shop
*** PASS: test_cases/q3/select_shop2.test
***     shopSmart(order, shops) selects the cheapest shop
Welcome to shop1 fruit shop
Welcome to shop2 fruit shop
Welcome to shop3 fruit shop
*** PASS: test_cases/q3/select_shop3.test
***     shopSmart(order, shops) selects the cheapest shop

### Question q3: 10/10 ###
```