C950: Task 2

A. Develop a hash table, without using any additional libraries or classes, that has an insertion function that takes the package ID as input and inserts each of the following data components into the hash table:
- delivery address
- delivery deadline
- delivery city
- delivery zip code
- package weight
- delivery status (i.e., at the hub, en route, or delivered), including the delivery time

The Hash Table has an insert function, it stores all the given components into a tuple. The package ID is the key to this bucket.

```
def insert (self,key):
    key = int(key)
```

```
bd = self.hashPackageData[bucketID]
bucketPackage = (bd.address, bd.deadline, bd.city, bd.zipCode, bd.weight, bd.status)
```

```
self.hashMap[bucketID] = Bucket(key, bucketPackage)
```

B. Develop a look-up function that takes the package ID as input and returns *each* of the following corresponding data components:
- delivery address
- delivery deadline
- delivery city
- delivery zip code
- package weight
- delivery status (i.e., at the hub, en route, or delivered), including the delivery time

The find() function uses the package ID as the key to retrieve all of the contents of that package which were stored into a bucket with a key that is equal to the package ID located in the Hash Table at an index that is key-1 (found using hash() function).

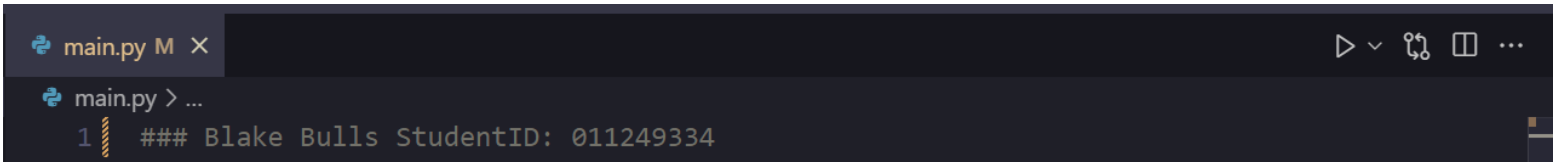
```
def find(self, key):
    key = int(key)
```

```
bucketID = self.hash(key)
bucket = self.hashMap[bucketID]
```

```
return str(bucket)
```

C. Write an original program that will deliver *all* packages and meet all requirements using the attached supporting documents "Salt Lake City Downtown Map," "WGUPS Distance Table," and "WGUPS Package File."

1. Create an identifying comment within the first line of a file named "main.py" that includes your student ID.



2. Include comments in your code to explain both the process and the flow of the program.

Extensive comments are provided.

D. Provide an intuitive interface for the user to view the delivery status (including the delivery time) of any package at any time and the total mileage traveled by all trucks. (The delivery status should report the package as at the hub, en route, or delivered. Delivery status must include the time.)

Interface requests for the time from the user. And then asks what the user wants to do. If the user chooses Look up a Package, the interface will request a package ID. If the user chooses View All Packages, the interface will print all packages and then the time, total miles, and the verification messages that packages met all specifications.

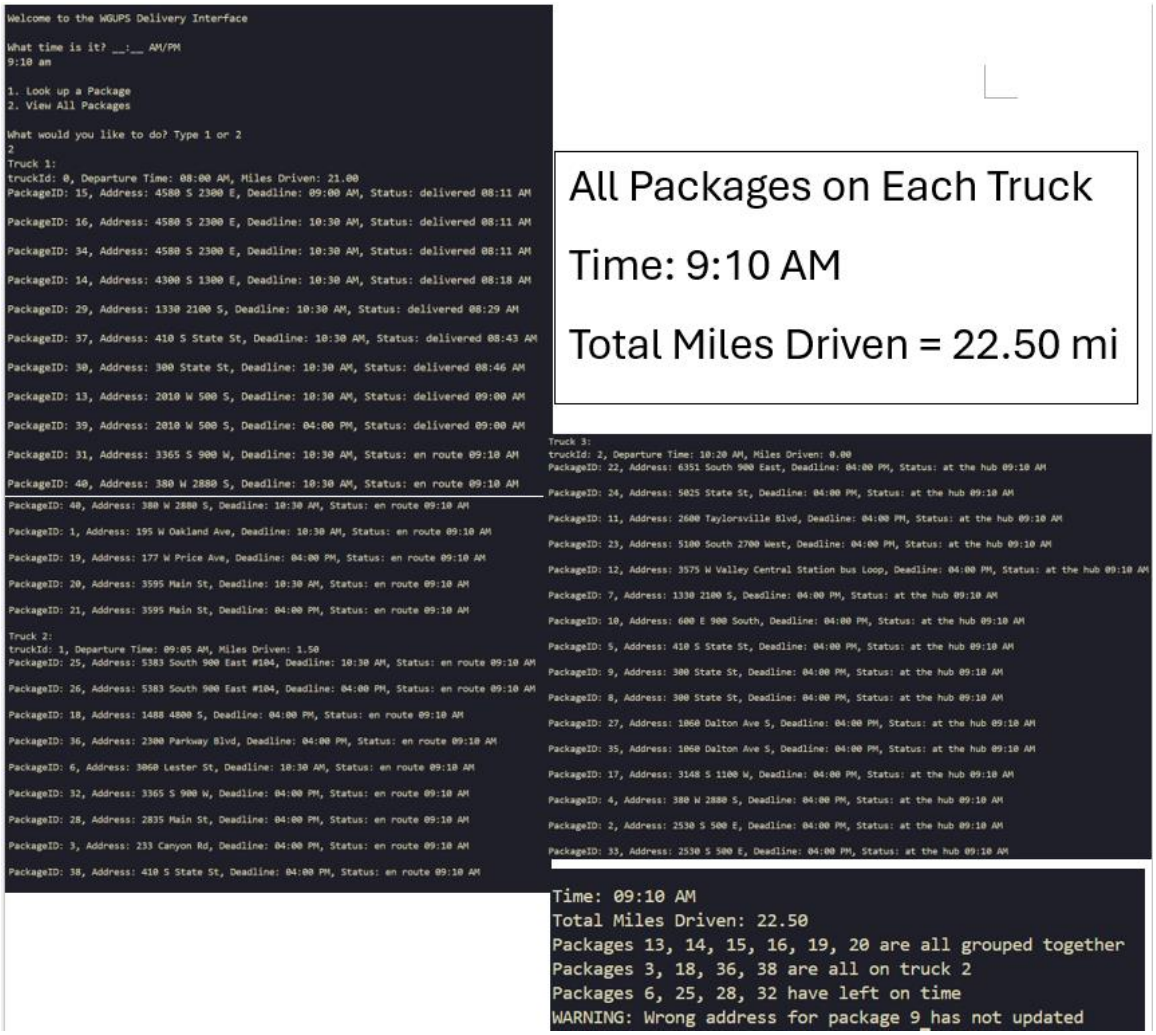
```
Welcome to the WGUPS Delivery Interface

What time is it? __:__ AM/PM
```

```
1. Look up a Package
2. View All Packages

What would you like to do? Type 1 or 2
```

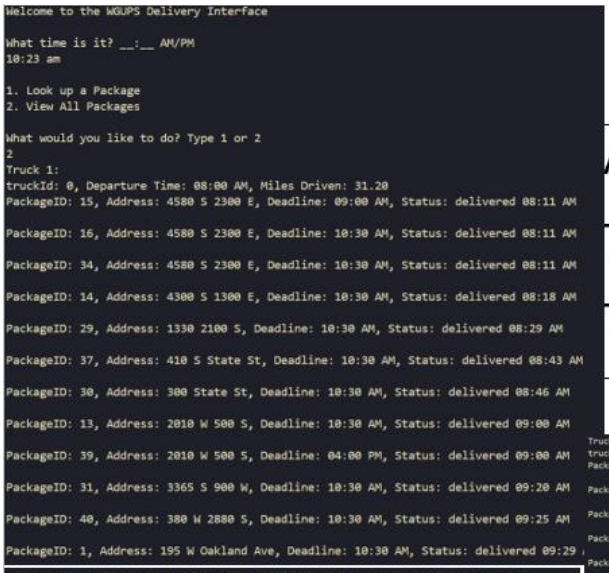
```
What would you like to do? Type 1 or 2
1

Checking the status of a package.....
What is the packageID that you would like to check?
```

```
Time: 04:00 PM
Total Miles Driven: 108.40
Packages 13, 14, 15, 16, 19, 20 are all grouped together
Packages 3, 18, 36, 38 are all on truck 2
Packages 6, 25, 28, 32 have left on time
Wrong address for package 9 has been corrected
```

1. Provide screenshots to show the status of *all* packages loaded onto *each* truck at a time between 8:35 a.m. and 9:25 a.m.



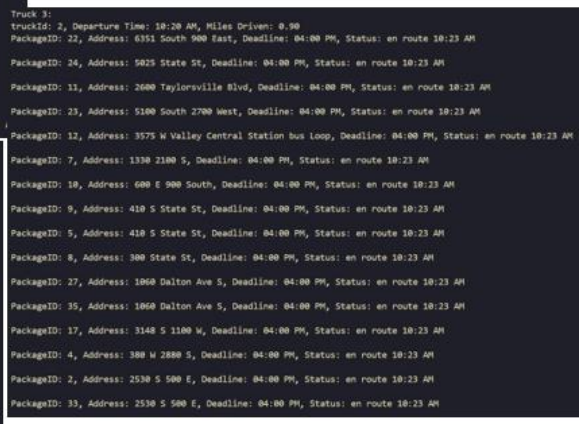
2. Provide screenshots to show the status of *all* packages loaded onto *each* truck at a time between 9:35 a.m. and 10:25 a.m.

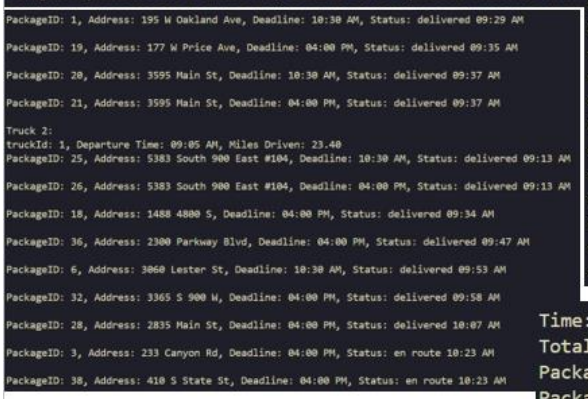


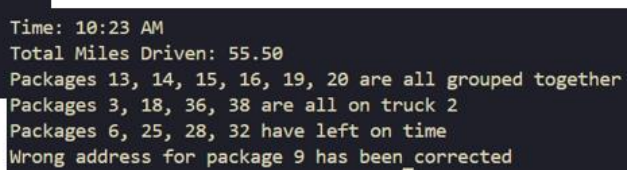
All Packages on Each Truck

Time: 10:23 AM

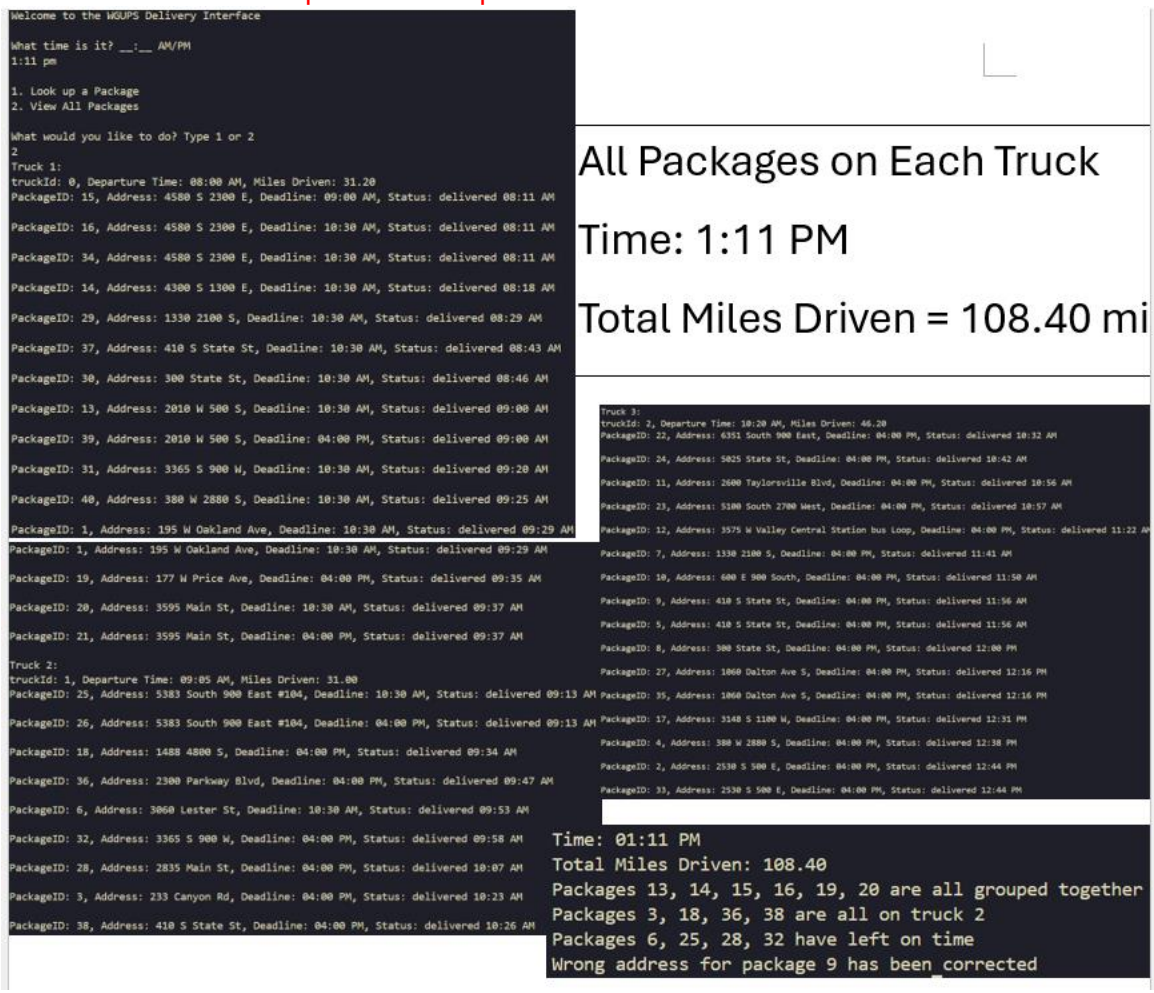
Total Miles Driven = 55.50 mi







3. Provide screenshots to show the status of *all* packages loaded onto *each* truck at a time between 12:03 p.m. and 1:12 p.m.

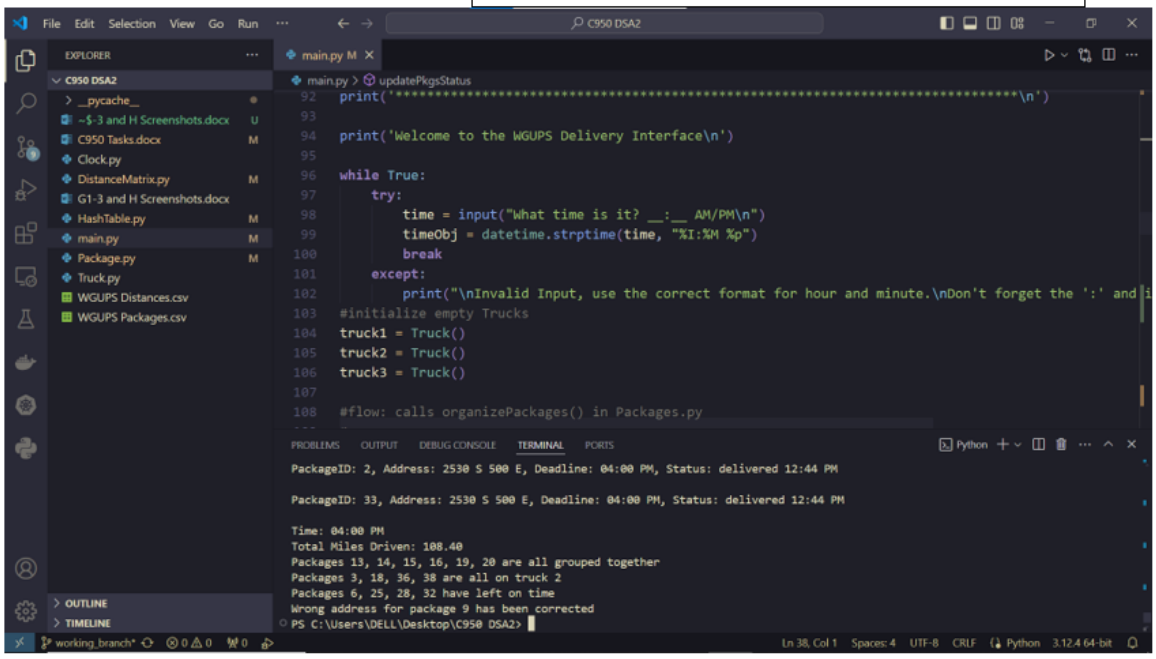


E. Provide screenshots showing successful completion of the code that includes the total mileage traveled by *all* trucks.

Code Execution

Time: 4:00pm aka EOD

Total Miles Driven = 108.40 mi



F. Justify the package delivery algorithm used in the solution as written in the original program by doing the following:

1. Describe **two or more** strengths of the algorithm used in the solution.

Strengths of bestTour algorithm: It adapts its result depending on the deadline of the packages. It provides optimization of the entire tour used by the truck to deliver its packages by continuing its iterations as long as an improved tour is discovered.

2. Verify that the algorithm used in the solution meets *all* requirements in the scenario.

The algorithm meets all required specifications.

```
Time: 01:11 PM
Total Miles Driven: 108.40
Packages 13, 14, 15, 16, 19, 20 are all grouped together
Packages 3, 18, 36, 38 are all on truck 2
Packages 6, 25, 28, 32 have left on time
Wrong address for package 9 has been corrected
```

3. Identify **two** other named algorithms that are different from the algorithm implemented in the solution and would meet *all* requirements in the scenario.
   a. Describe how *both* algorithms identified in part F3 are different from the algorithm used in the solution.

Dijkstra's algorithm or a genetic algorithm could have been used to meet the scenario's requirements

Dijkstra's algorithm is used for finding the shortest path in weighted directed or undirected graph. It will provide the exact shortest path. Although both are deterministic, meaning they will continue searching as long as they find an improvement, two-opt differs in that it is a heuristic algorithm. Meaning, two-opt will find approximate solutions that aren't guaranteed to be as precise as Dijkstra's algorithm (Navone, 2020).

A genetic algorithm has a population of potential solutions that incorporate operators to "evolve" the population with an element of randomness over generations. These generations can be evaluated based on the quality of the solution provided. This is very different from two-opt, which simply checks the distance of two edges and keeps the shortest one. Genetic algorithms can also have lots of variety in their solutions, whereas two-opt iteratively makes small local changes that lack variety (MathWorks, 2019).

G. Describe what you would do differently, other than the two algorithms identified in part F3, if you did this project again, including details of the modifications that would be made.

One aspect that I would do differently is the distance matrix design. As it is, it seems awkward in the way it chains together multiple functions. Perhaps a dictionary of addresses as keys and tuples of neighboring addresses and their distances would have been a more concise and less space consuming option.

H. Verify that the data structure used in the solution meets *all* requirements in the scenario.

The data structure used meets all requirements in the scenario.

1. Identify **two** other data structures that could meet the same requirements in the scenario.
   a. Describe how *each* data structure identified in H1 is different from the data structure used in the solution.

A binary search tree or a trie are 2 possible data structures that could be used.

A binary search tree guarantees that all operations will work in O(log n), whereas with hashing, some operations can be O(n^2). Binary search trees require less memory and computing (Advantages of BST over Hash Table, 2015).

A trie, like a hash table, prioritizes quick retrieval of data. This seems to be a likely choice for this project due to the frequency of looking up different packages for their status. Searching for the existence of a particular string like " wrong address" or "truck 2" would have a much faster time complexity than in the current hash table implementation (Hash Table vs Trie, 2022).

I. Acknowledge sources, using in-text citations and references, for content that is quoted, paraphrased, or summarized.

2-opt algorithm to solve the Travelling Salesman Problem in Python. (n.d.). Stack Overflow. https://stackoverflow.com/questions/53275314/2-opt-algorithm-to-solve-the-travelling-salesman-problem-in-python

Advantages of BST over Hash Table. (2015, April 1). GeeksforGeeks. https://www.geeksforgeeks.org/advantages-of-bst-over-hash-table/

Create an incremental ID in a Python class with subclasses each maintaining their own ID series. (n.d.). Stack Overflow. Retrieved July 22, 2024, from https://stackoverflow.com/questions/71520394/create-an-incremental-id-in-a-python-class-with-subclasses-each-maintaining-thei

Hash Table vs Trie. (2022, December 15). GeeksforGeeks. https://www.geeksforgeeks.org/hash-table-vs-trie/

MathWorks. (2019). *What Is the Genetic Algorithm?* Mathworks.com. https://www.mathworks.com/help/gads/what-is-the-genetic-algorithm.html

Navone, E. C. (2020, September 28). *Dijkstra's Shortest Path Algorithm - A Detailed and Visual Introduction*. FreeCodeCamp.org. https://www.freecodecamp.org/news/dijkstras-shortest-path-algorithm-visual-introduction/

Oggi AI - Artificial Intelligence Today. (2016, April 16). *Python: 2 Ways to Represent GRAPHS*. YouTube. https://www.youtube.com/watch?v=HDUzBEG1GlA&t=486s

PageKey. (2017, November 26). *Hash Tables in Python*. YouTube. https://www.youtube.com/watch?v=zHi5v78W1f0&t=628s

J. Demonstrate professional communication in the content and presentation of your submission.

Extensive communication is used.