C950 Task-2 WGUPS Write-Up

Brandon Robinson

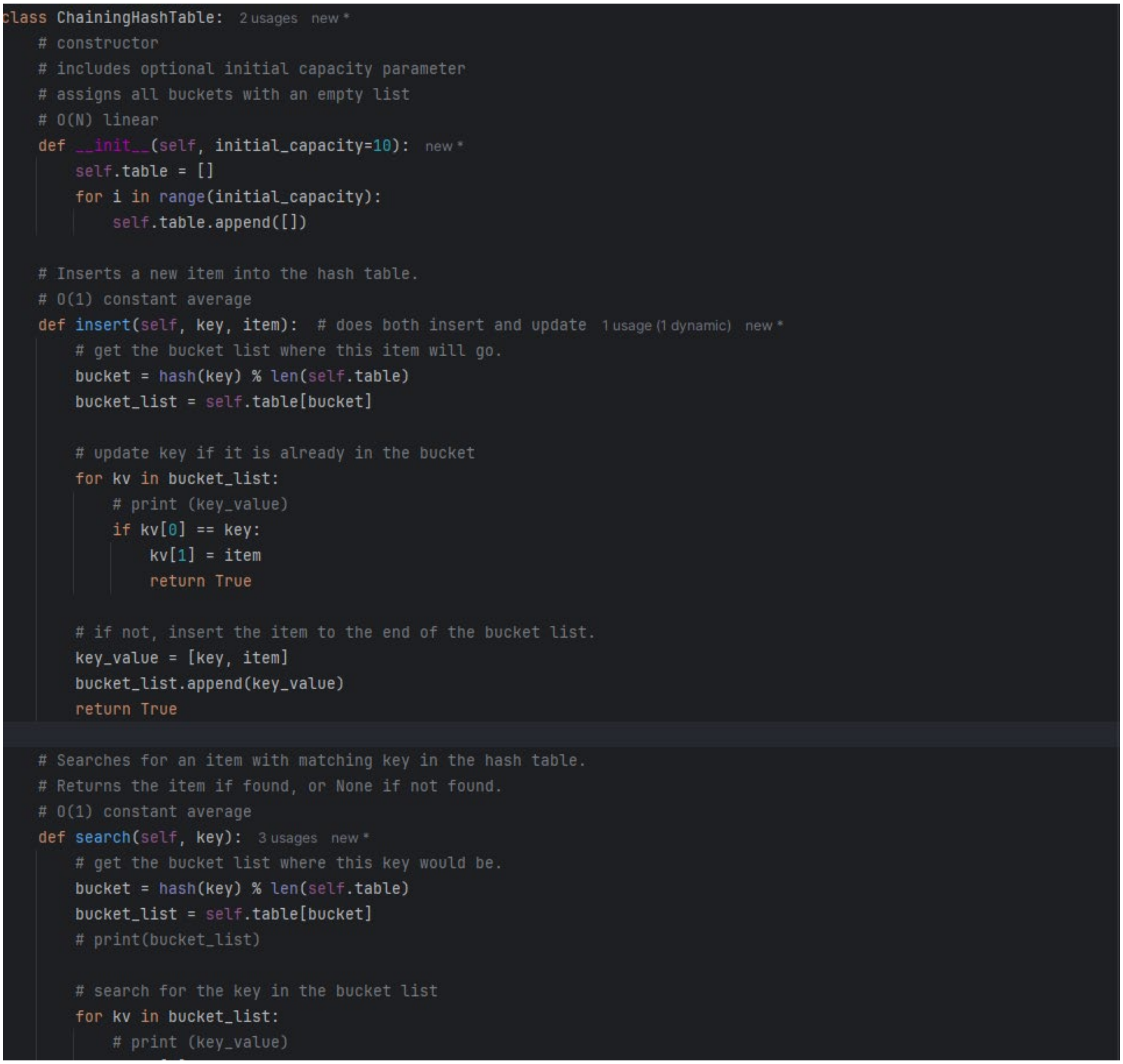
ID #006868740

WGU Email: brob538@wgu.edu

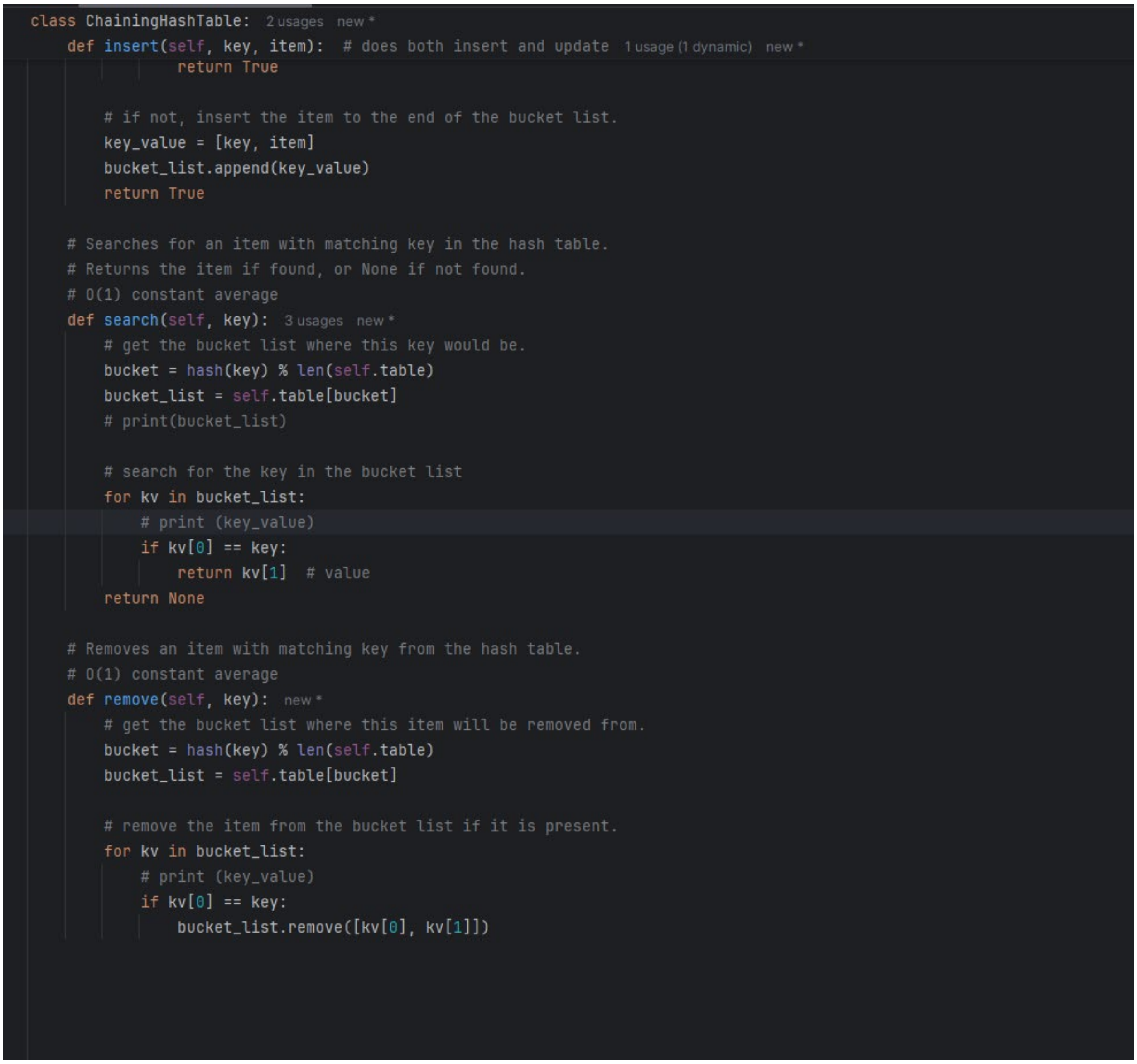
01/24/25

C950 Data Structures and Algorithms II

A. Hash Table

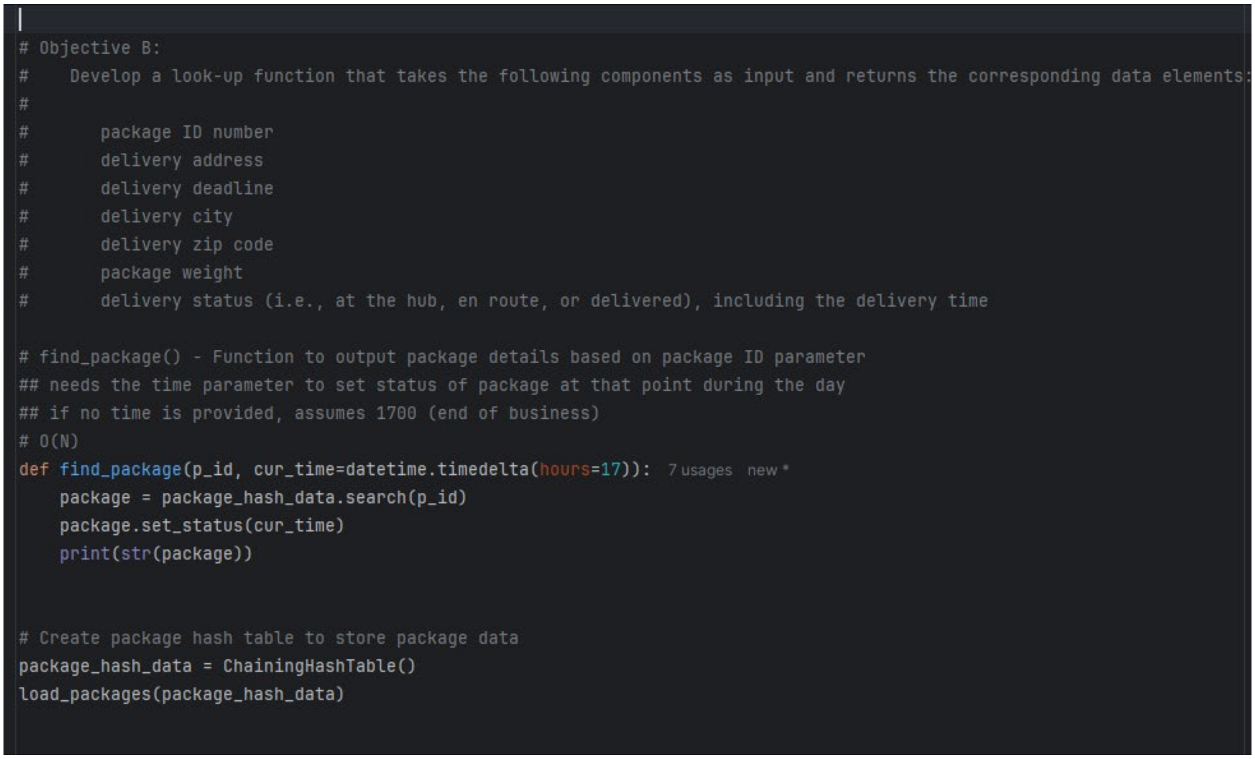


```
class ChainingHashTable:  2 usages  new *
    # constructor
    # includes optional initial capacity parameter
    # assigns all buckets with an empty list
    # O(N) linear
    def __init__(self, initial_capacity=10):  new *
        self.table = []
        for i in range(initial_capacity):
            self.table.append([])

    # Inserts a new item into the hash table.
    # O(1) constant average
    def insert(self, key, item):  # does both insert and update  1 usage (1 dynamic)  new *
        # get the bucket list where this item will go.
        bucket = hash(key) % len(self.table)
        bucket_list = self.table[bucket]

        # update key if it is already in the bucket
        for kv in bucket_list:
            # print (key_value)
            if kv[0] == key:
                kv[1] = item
                return True

        # if not, insert the item to the end of the bucket list.
        key_value = [key, item]
        bucket_list.append(key_value)
        return True

    # Searches for an item with matching key in the hash table.
    # Returns the item if found, or None if not found.
    # O(1) constant average
    def search(self, key):  3 usages  new *
        # get the bucket list where this key would be.
        bucket = hash(key) % len(self.table)
        bucket_list = self.table[bucket]
        # print(bucket_list)

        # search for the key in the bucket list
        for kv in bucket_list:
            # print (key_value)
```



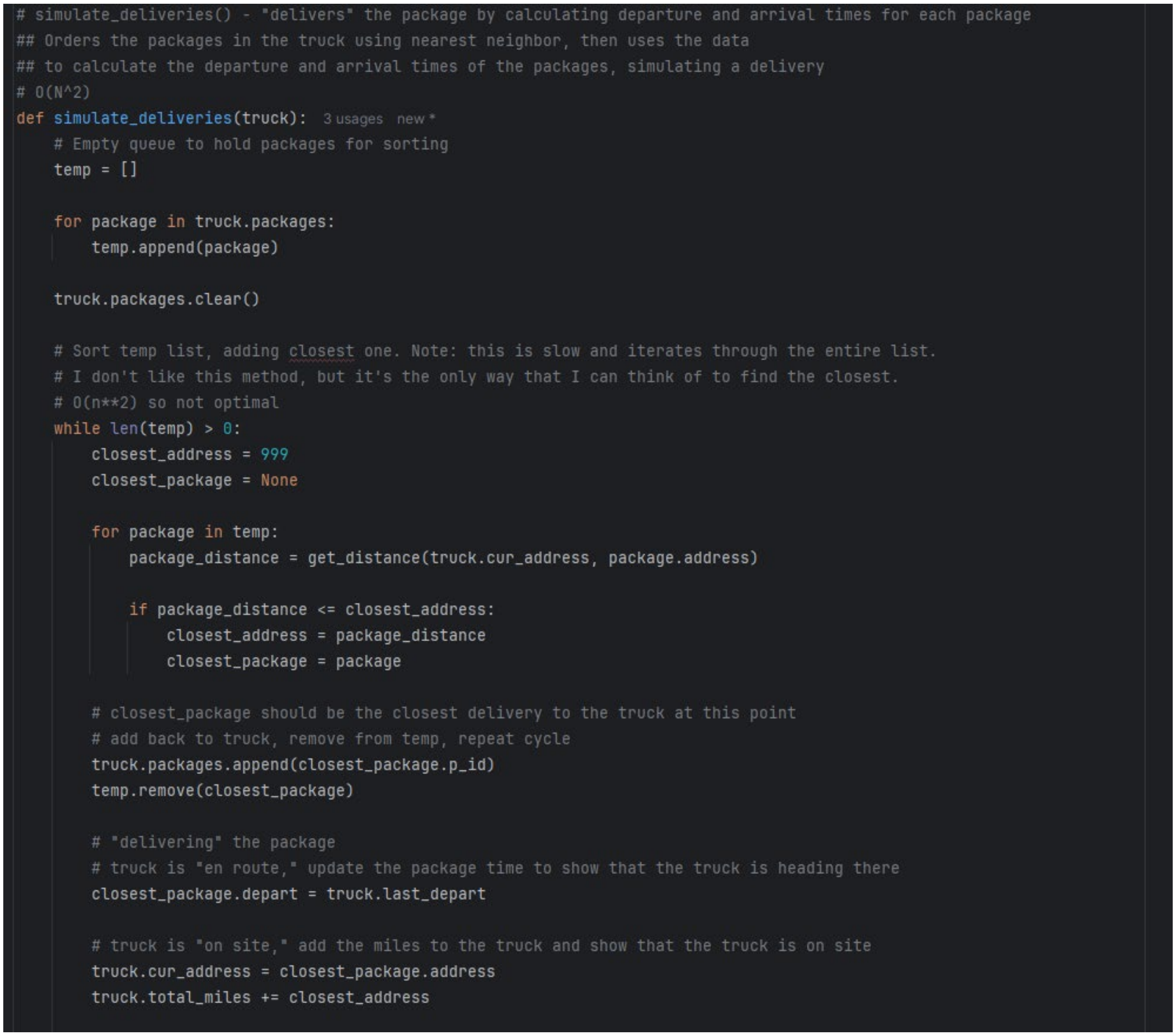
```
class ChainingHashTable:  2 usages  new *
    def insert(self, key, item):  # does both insert and update  1 usage (1 dynamic)  new *
                return True

        # if not, insert the item to the end of the bucket list.
        key_value = [key, item]
        bucket_list.append(key_value)
        return True

    # Searches for an item with matching key in the hash table.
    # Returns the item if found, or None if not found.
    # O(1) constant average
    def search(self, key):  3 usages  new *
        # get the bucket list where this key would be.
        bucket = hash(key) % len(self.table)
        bucket_list = self.table[bucket]
        # print(bucket_list)

        # search for the key in the bucket list
        for kv in bucket_list:
            # print (key_value)
            if kv[0] == key:
                return kv[1]  # value
        return None

    # Removes an item with matching key from the hash table.
    # O(1) constant average
    def remove(self, key):  new *
        # get the bucket list where this item will be removed from.
        bucket = hash(key) % len(self.table)
        bucket_list = self.table[bucket]

        # remove the item from the bucket list if it is present.
        for kv in bucket_list:
            # print (key_value)
            if kv[0] == key:
                bucket_list.remove([kv[0], kv[1]])
```

B. Look-Up Functions



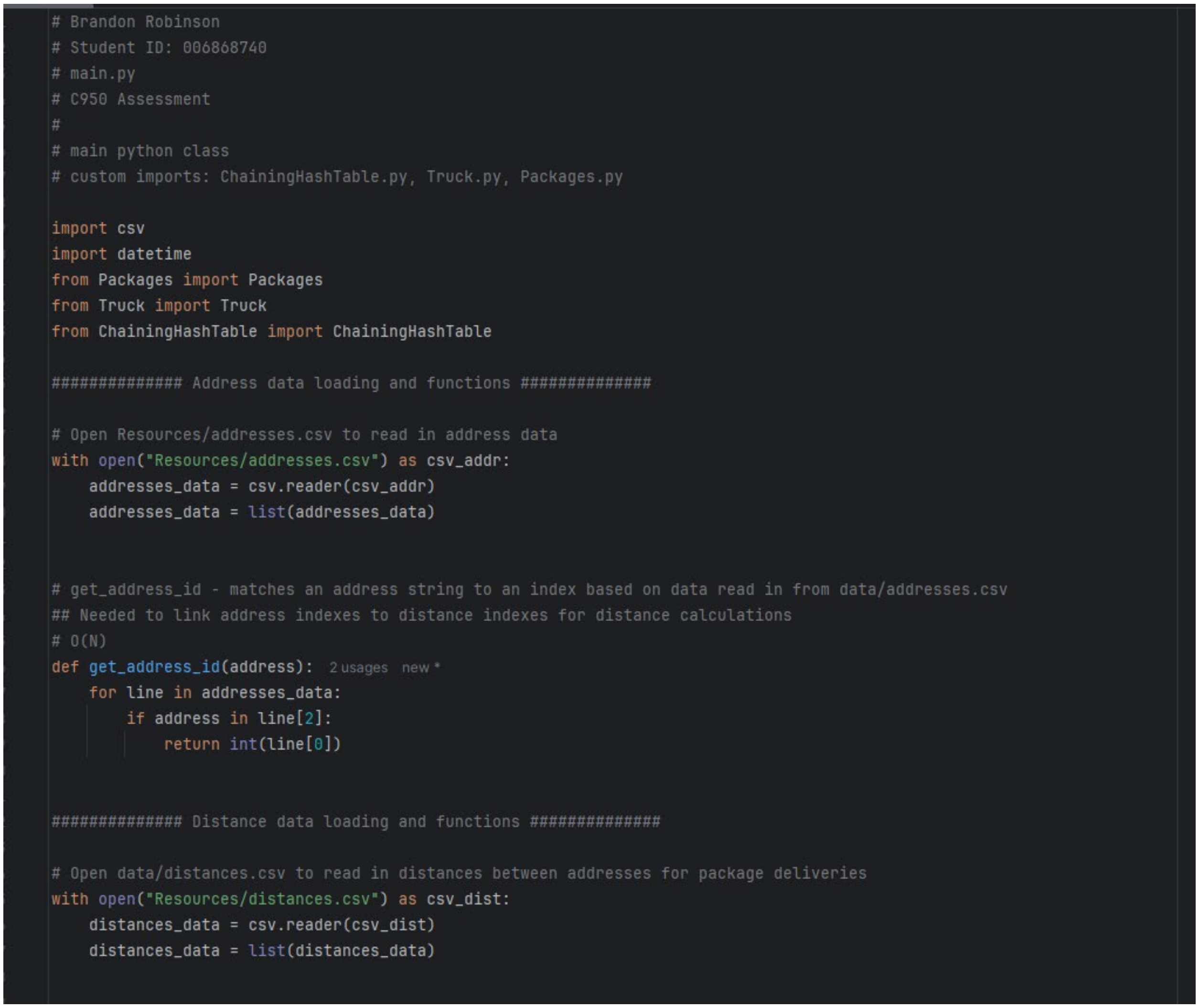
```
# Objective B:
#    Develop a look-up function that takes the following components as input and returns the corresponding data elements:
#
#       package ID number
#       delivery address
#       delivery deadline
#       delivery city
#       delivery zip code
#       package weight
#       delivery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

# find_package() - Function to output package details based on package ID parameter
## needs the time parameter to set status of package at that point during the day
## if no time is provided, assumes 1700 (end of business)
# O(N)
def find_package(p_id, cur_time=datetime.timedelta(hours=17)):  7 usages  new *
    package = package_hash_data.search(p_id)
    package.set_status(cur_time)
    print(str(package))


# Create package hash table to store package data
package_hash_data = ChainingHashTable()
load_packages(package_hash_data)
```

C. Original Code



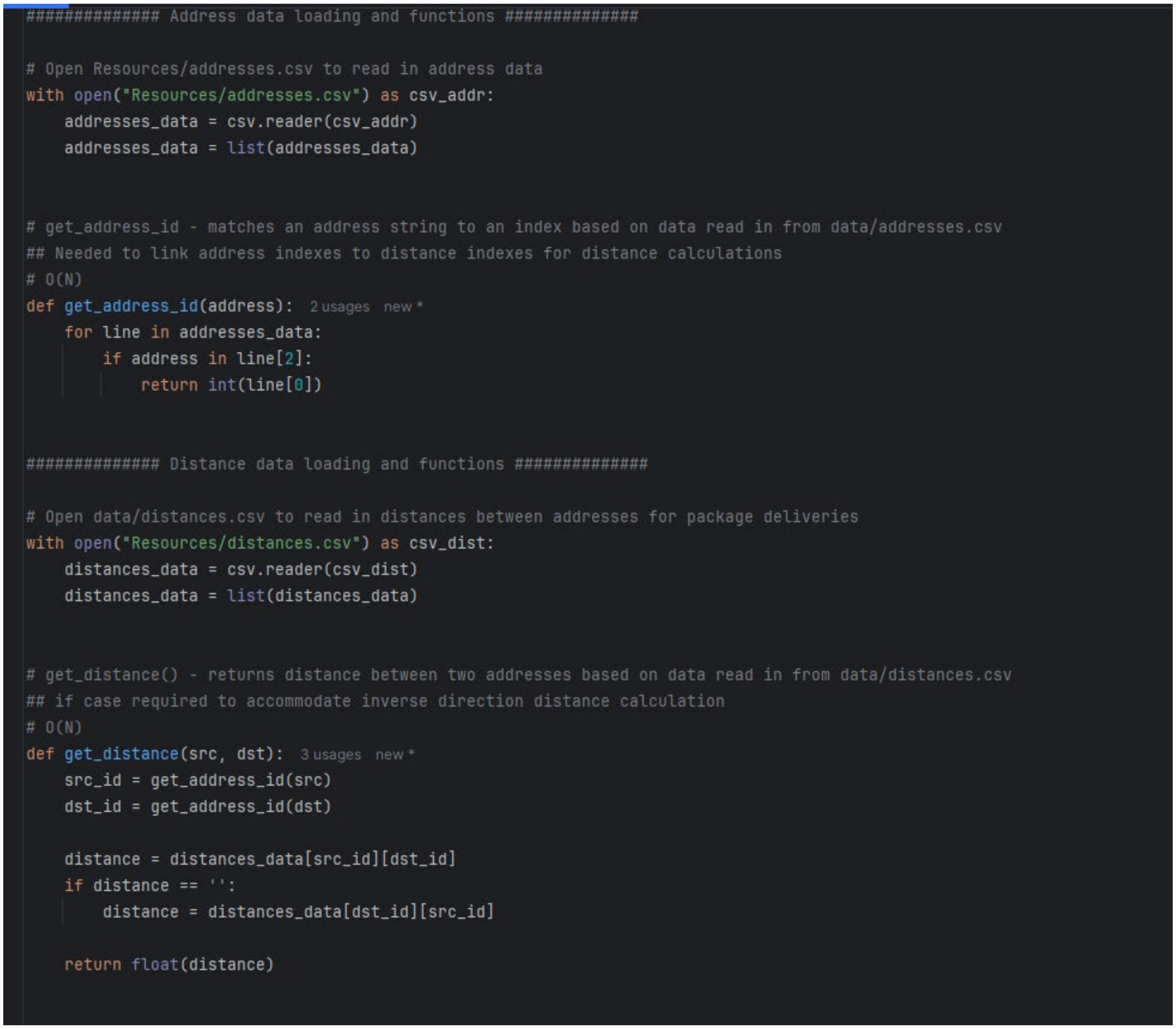
```python
# simulate_deliveries() - "delivers" the package by calculating departure and arrival times for each package
## Orders the packages in the truck using nearest neighbor, then uses the data
## to calculate the departure and arrival times of the packages, simulating a delivery
# O(N^2)
def simulate_deliveries(truck):
    # Empty queue to hold packages for sorting
    temp = []

    for package in truck.packages:
        temp.append(package)

    truck.packages.clear()

    # Sort temp list, adding closest one. Note: this is slow and iterates through the entire list.
    # I don't like this method, but it's the only way that I can think of to find the closest.
    # O(n**2) so not optimal
    while len(temp) > 0:
        closest_address = 999
        closest_package = None

        for package in temp:
            package_distance = get_distance(truck.cur_address, package.address)

            if package_distance <= closest_address:
                closest_address = package_distance
                closest_package = package

        # closest_package should be the closest delivery to the truck at this point
        # add back to truck, remove from temp, repeat cycle
        truck.packages.append(closest_package.p_id)
        temp.remove(closest_package)

        # "delivering" the package
        # truck is "en route," update the package time to show that the truck is heading there
        closest_package.depart = truck.last_depart

        # truck is "on site," add the miles to the truck and show that the truck is on site
        truck.cur_address = closest_package.address
        truck.total_miles += closest_address
```

C1. Identification Information

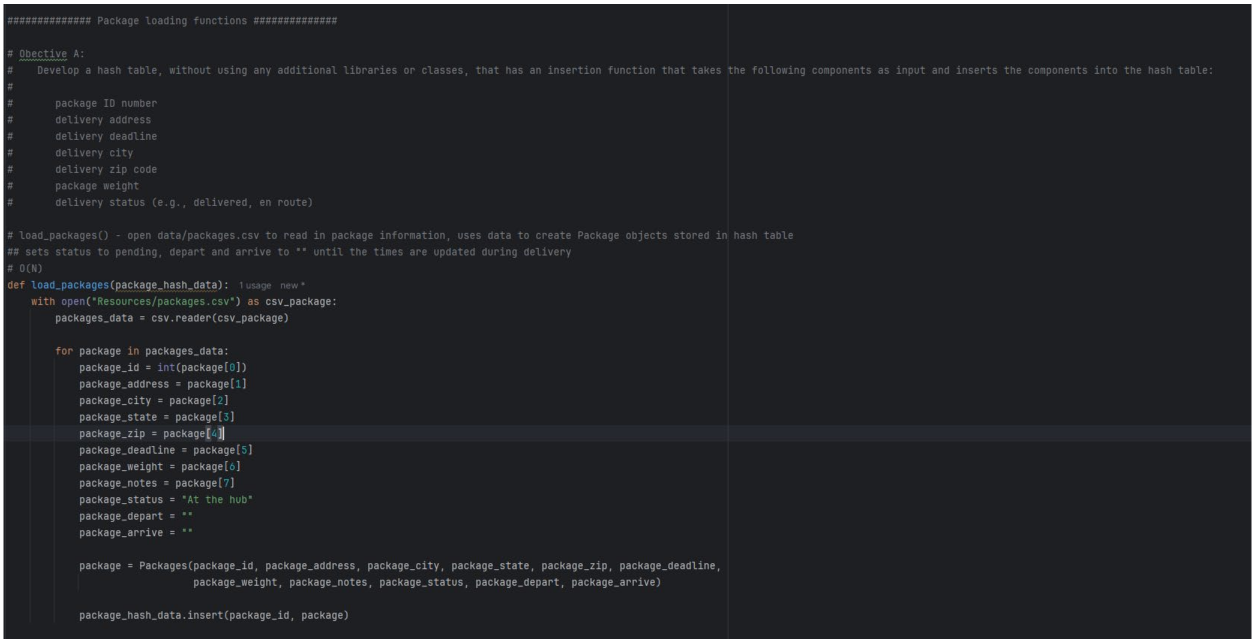


```
# Brandon Robinson
# Student ID: 006868740
# main.py
# C950 Assessment
#
# main python class
# custom imports: ChainingHashTable.py, Truck.py, Packages.py

import csv
import datetime
from Packages import Packages
from Truck import Truck
from ChainingHashTable import ChainingHashTable

############## Address data loading and functions ##############

# Open Resources/addresses.csv to read in address data
with open("Resources/addresses.csv") as csv_addr:
    addresses_data = csv.reader(csv_addr)
    addresses_data = list(addresses_data)


# get_address_id - matches an address string to an index based on data read in from data/addresses.csv
## Needed to link address indexes to distance indexes for distance calculations
# O(N)
def get_address_id(address):  2 usages  new *
    for line in addresses_data:
        if address in line[2]:
            return int(line[0])


############## Distance data loading and functions ##############

# Open data/distances.csv to read in distances between addresses for package deliveries
with open("Resources/distances.csv") as csv_dist:
    distances_data = csv.reader(csv_dist)
    distances_data = list(distances_data)
```

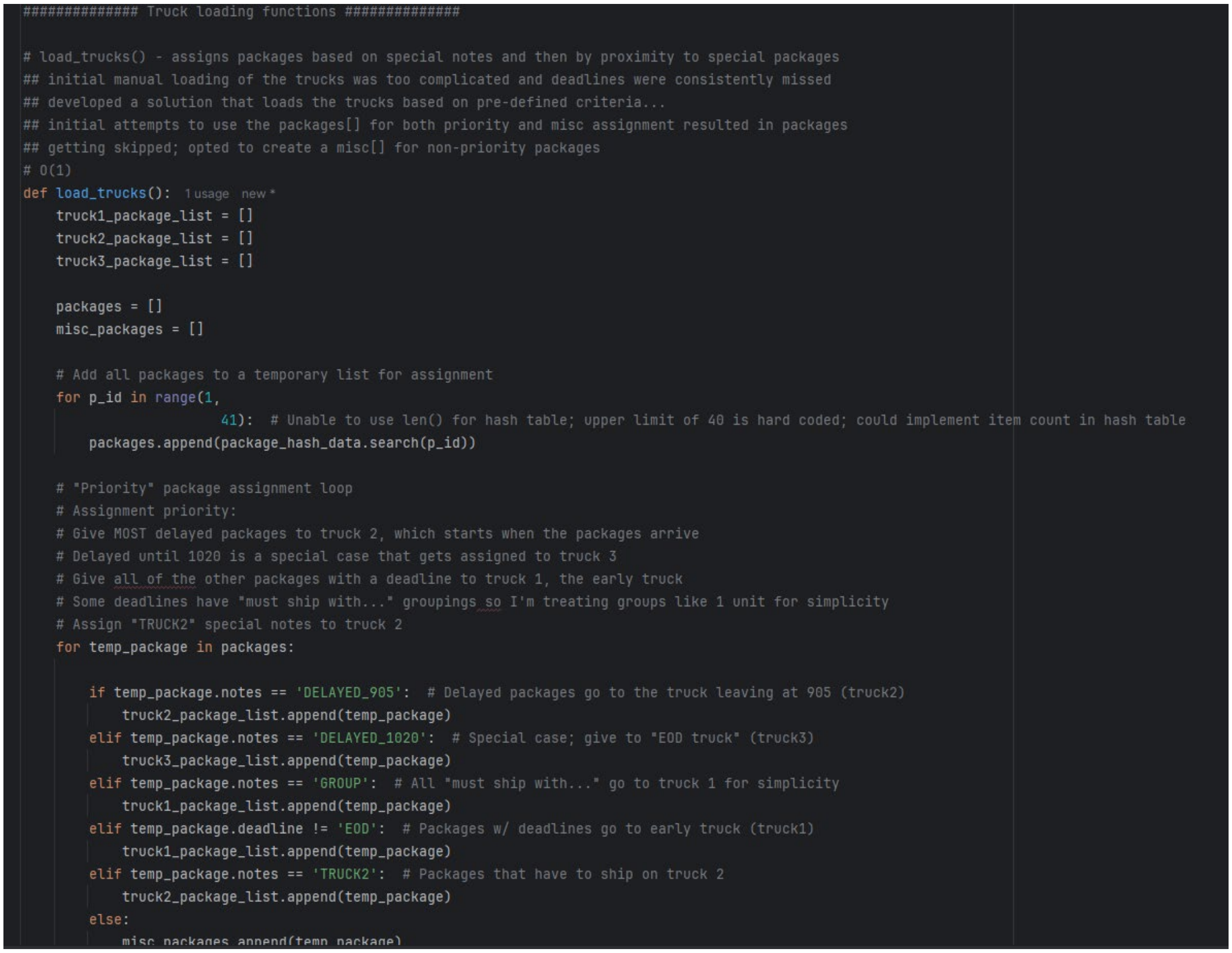
C2. Process and Flow Comments



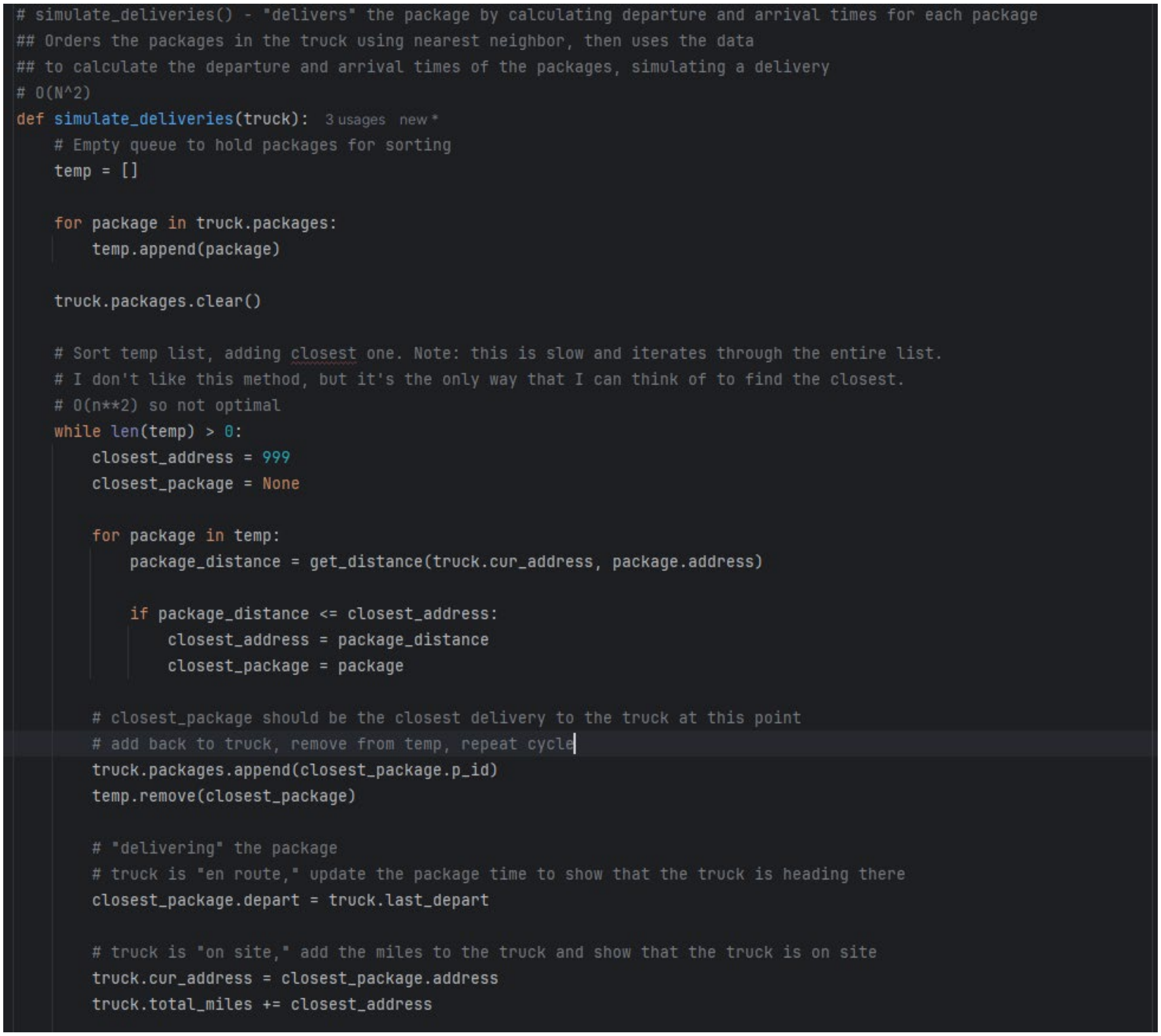
```
############## Address data loading and functions ##############

# Open Resources/addresses.csv to read in address data
with open("Resources/addresses.csv") as csv_addr:
    addresses_data = csv.reader(csv_addr)
    addresses_data = list(addresses_data)


# get_address_id - matches an address string to an index based on data read in from data/addresses.csv
## Needed to link address indexes to distance indexes for distance calculations
# O(N)
def get_address_id(address):
    for line in addresses_data:
        if address in line[2]:
            return int(line[0])


############## Distance data loading and functions ##############

# Open data/distances.csv to read in distances between addresses for package deliveries
with open("Resources/distances.csv") as csv_dist:
    distances_data = csv.reader(csv_dist)
    distances_data = list(distances_data)


# get_distance() - returns distance between two addresses based on data read in from data/distances.csv
## if case required to accommodate inverse direction distance calculation
# O(N)
def get_distance(src, dst):
    src_id = get_address_id(src)
    dst_id = get_address_id(dst)

    distance = distances_data[src_id][dst_id]
    if distance == '':
        distance = distances_data[dst_id][src_id]

    return float(distance)
```



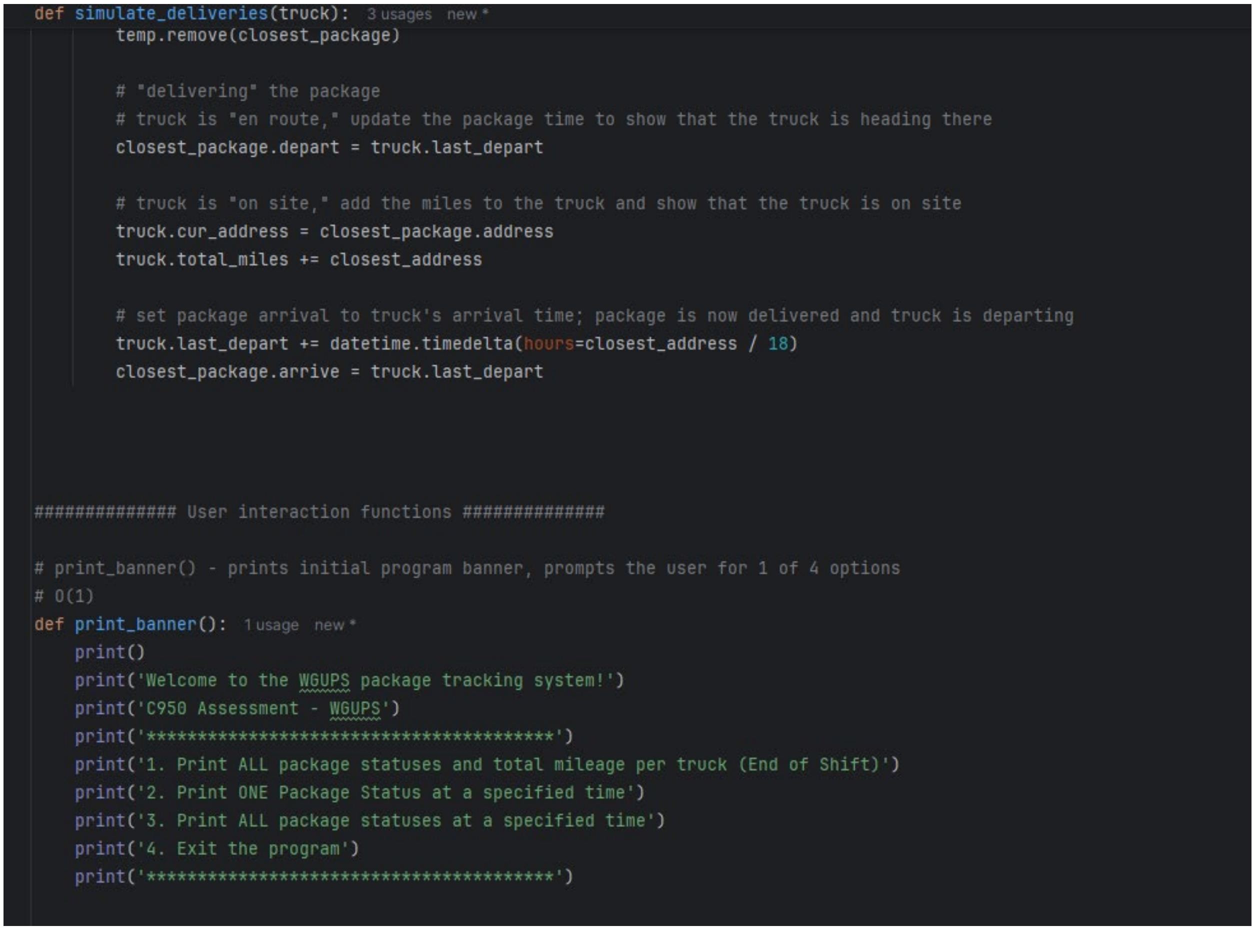
```
############## Package loading functions ##############

# Obective A:
#    Develop a hash table, without using any additional libraries or classes, that has an insertion function that takes the following components as input and inserts the components into the hash table:
#
#        package ID number
#        delivery address
#        delivery deadline
#        delivery city
#        delivery zip code
#        package weight
#        delivery status (e.g., delivered, en route)

# load_packages() - open data/packages.csv to read in package information, uses data to create Package objects stored in hash table
## sets status to pending, depart and arrive to "" until the times are updated during delivery
# O(N)
def load_packages(package_hash_data):  1 usage  new *
    with open("Resources/packages.csv") as csv_package:
        packages_data = csv.reader(csv_package)

        for package in packages_data:
            package_id = int(package[0])
            package_address = package[1]
            package_city = package[2]
            package_state = package[3]
            package_zip = package[4]
            package_deadline = package[5]
            package_weight = package[6]
            package_notes = package[7]
            package_status = "At the hub"
            package_depart = ""
            package_arrive = ""

            package = Packages(package_id, package_address, package_city, package_state, package_zip, package_deadline,
                               package_weight, package_notes, package_status, package_depart, package_arrive)

            package_hash_data.insert(package_id, package)
```



```
############## Truck loading functions ##############

# load_trucks() - assigns packages based on special notes and then by proximity to special packages
## initial manual loading of the trucks was too complicated and deadlines were consistently missed
## developed a solution that loads the trucks based on pre-defined criteria...
## initial attempts to use the packages[] for both priority and misc assignment resulted in packages
## getting skipped; opted to create a misc[] for non-priority packages
# O(1)
def load_trucks():  1 usage  new *
    truck1_package_list = []
    truck2_package_list = []
    truck3_package_list = []

    packages = []
    misc_packages = []

    # Add all packages to a temporary list for assignment
    for p_id in range(1,
                      41):  # Unable to use len() for hash table; upper limit of 40 is hard coded; could implement item count in hash table
        packages.append(package_hash_data.search(p_id))

    # "Priority" package assignment loop
    # Assignment priority:
    # Give MOST delayed packages to truck 2, which starts when the packages arrive
    # Delayed until 1020 is a special case that gets assigned to truck 3
    # Give all of the other packages with a deadline to truck 1, the early truck
    # Some deadlines have "must ship with..." groupings so I'm treating groups like 1 unit for simplicity
    # Assign "TRUCK2" special notes to truck 2
    for temp_package in packages:

        if temp_package.notes == 'DELAYED_905':  # Delayed packages go to the truck leaving at 905 (truck2)
            truck2_package_list.append(temp_package)
        elif temp_package.notes == 'DELAYED_1020':  # Special case; give to "EOD truck" (truck3)
            truck3_package_list.append(temp_package)
        elif temp_package.notes == 'GROUP':  # All "must ship with..." go to truck 1 for simplicity
            truck1_package_list.append(temp_package)
        elif temp_package.deadline != 'EOD':  # Packages w/ deadlines go to early truck (truck1)
            truck1_package_list.append(temp_package)
        elif temp_package.notes == 'TRUCK2':  # Packages that have to ship on truck 2
            truck2_package_list.append(temp_package)
        else:
            misc_packages.append(temp_package)
```

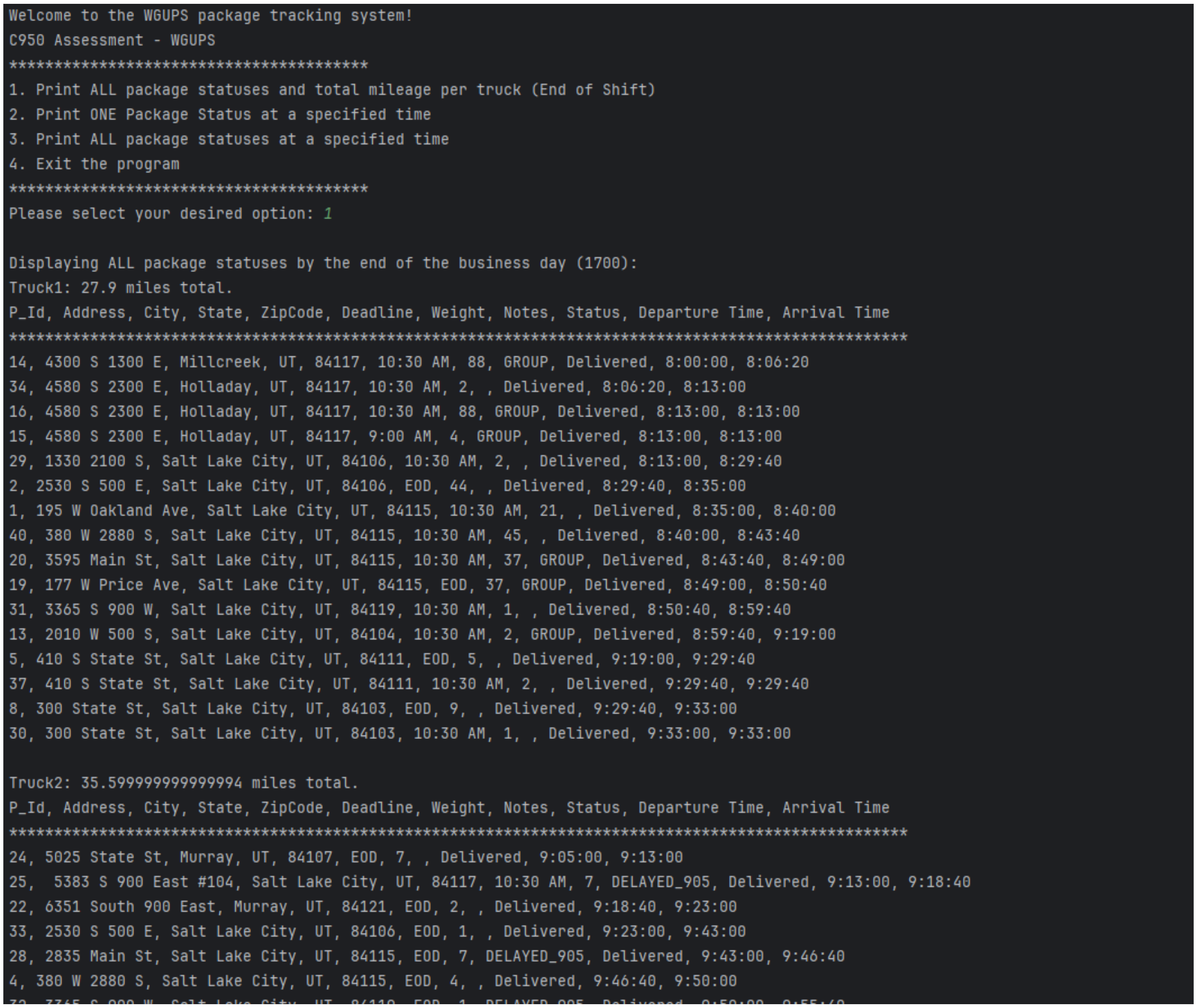


```
# simulate_deliveries() - "delivers" the package by calculating departure and arrival times for each package
## Orders the packages in the truck using nearest neighbor, then uses the data
## to calculate the departure and arrival times of the packages, simulating a delivery
# O(N^2)
def simulate_deliveries(truck):
    # Empty queue to hold packages for sorting
    temp = []

    for package in truck.packages:
        temp.append(package)

    truck.packages.clear()

    # Sort temp list, adding closest one. Note: this is slow and iterates through the entire list.
    # I don't like this method, but it's the only way that I can think of to find the closest.
    # O(n**2) so not optimal
    while len(temp) > 0:
        closest_address = 999
        closest_package = None

        for package in temp:
            package_distance = get_distance(truck.cur_address, package.address)

            if package_distance <= closest_address:
                closest_address = package_distance
                closest_package = package

        # closest_package should be the closest delivery to the truck at this point
        # add back to truck, remove from temp, repeat cycle
        truck.packages.append(closest_package.p_id)
        temp.remove(closest_package)

        # "delivering" the package
        # truck is "en route," update the package time to show that the truck is heading there
        closest_package.depart = truck.last_depart

        # truck is "on site," add the miles to the truck and show that the truck is on site
        truck.cur_address = closest_package.address
        truck.total_miles += closest_address
```

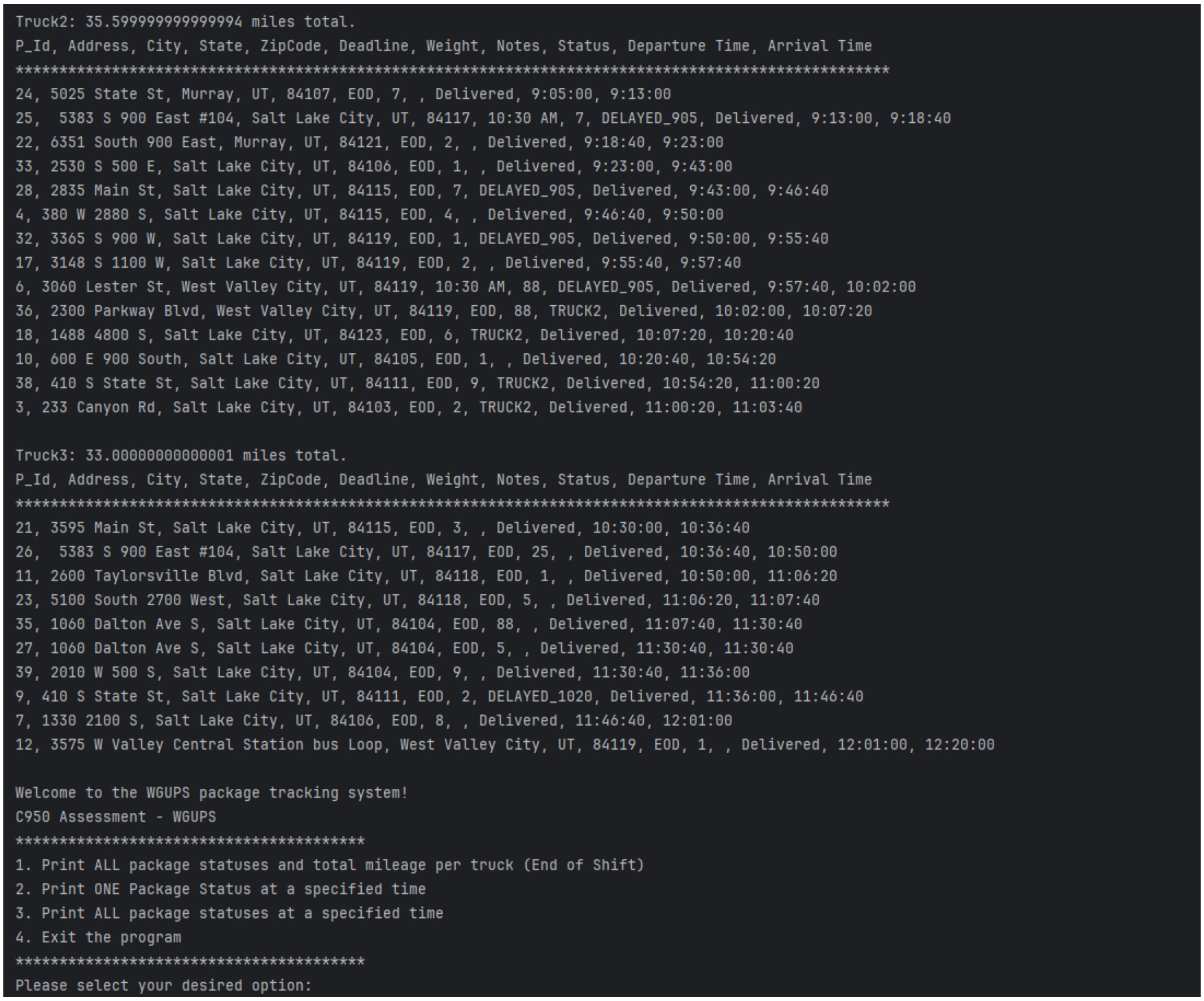


```
def simulate_deliveries(truck):
        temp.remove(closest_package)

        # "delivering" the package
        # truck is "en route," update the package time to show that the truck is heading there
        closest_package.depart = truck.last_depart

        # truck is "on site," add the miles to the truck and show that the truck is on site
        truck.cur_address = closest_package.address
        truck.total_miles += closest_address

        # set package arrival to truck's arrival time; package is now delivered and truck is departing
        truck.last_depart += datetime.timedelta(hours=closest_address / 18)
        closest_package.arrive = truck.last_depart




############## User interaction functions ##############

# print_banner() - prints initial program banner, prompts the user for 1 of 4 options
# O(1)
def print_banner():
    print()
    print('Welcome to the WGUPS package tracking system!')
    print('C950 Assessment - WGUPS')
    print('****************************************')
    print('1. Print ALL package statuses and total mileage per truck (End of Shift)')
    print('2. Print ONE Package Status at a specified time')
    print('3. Print ALL package statuses at a specified time')
    print('4. Exit the program')
    print('****************************************')
```

D. Interface

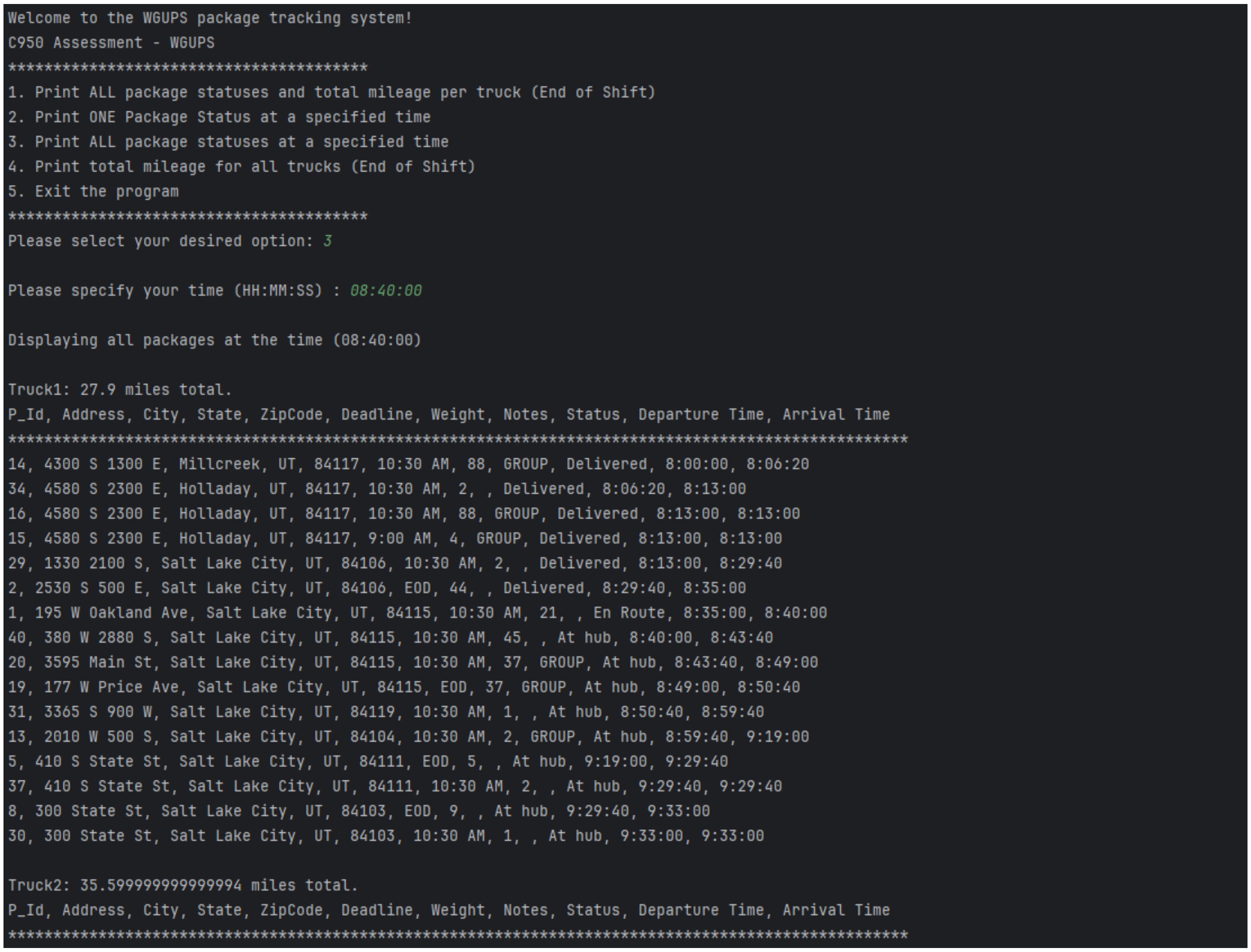


```
Welcome to the WGUPS package tracking system!
C950 Assessment - WGUPS
****************************************
1. Print ALL package statuses and total mileage per truck (End of Shift)
2. Print ONE Package Status at a specified time
3. Print ALL package statuses at a specified time
4. Exit the program
****************************************
Please select your desired option: 1

Displaying ALL package statuses by the end of the business day (1700):
Truck1: 27.9 miles total.
P_Id, Address, City, State, ZipCode, Deadline, Weight, Notes, Status, Departure Time, Arrival Time
****************************************************************************************************
14, 4300 S 1300 E, Millcreek, UT, 84117, 10:30 AM, 88, GROUP, Delivered, 8:00:00, 8:06:20
34, 4580 S 2300 E, Holladay, UT, 84117, 10:30 AM, 2, , Delivered, 8:06:20, 8:13:00
16, 4580 S 2300 E, Holladay, UT, 84117, 10:30 AM, 88, GROUP, Delivered, 8:13:00, 8:13:00
15, 4580 S 2300 E, Holladay, UT, 84117, 9:00 AM, 4, GROUP, Delivered, 8:13:00, 8:13:00
29, 1330 2100 S, Salt Lake City, UT, 84106, 10:30 AM, 2, , Delivered, 8:13:00, 8:29:40
2, 2530 S 500 E, Salt Lake City, UT, 84106, EOD, 44, , Delivered, 8:29:40, 8:35:00
1, 195 W Oakland Ave, Salt Lake City, UT, 84115, 10:30 AM, 21, , Delivered, 8:35:00, 8:40:00
40, 380 W 2880 S, Salt Lake City, UT, 84115, 10:30 AM, 45, , Delivered, 8:40:00, 8:43:40
20, 3595 Main St, Salt Lake City, UT, 84115, 10:30 AM, 37, GROUP, Delivered, 8:43:40, 8:49:00
19, 177 W Price Ave, Salt Lake City, UT, 84115, EOD, 37, GROUP, Delivered, 8:49:00, 8:50:40
31, 3365 S 900 W, Salt Lake City, UT, 84119, 10:30 AM, 1, , Delivered, 8:50:40, 8:59:40
13, 2010 W 500 S, Salt Lake City, UT, 84104, 10:30 AM, 2, GROUP, Delivered, 8:59:40, 9:19:00
5, 410 S State St, Salt Lake City, UT, 84111, EOD, 5, , Delivered, 9:19:00, 9:29:40
37, 410 S State St, Salt Lake City, UT, 84111, 10:30 AM, 2, , Delivered, 9:29:40, 9:29:40
8, 300 State St, Salt Lake City, UT, 84103, EOD, 9, , Delivered, 9:29:40, 9:33:00
30, 300 State St, Salt Lake City, UT, 84103, 10:30 AM, 1, , Delivered, 9:33:00, 9:33:00

Truck2: 35.599999999999994 miles total.
P_Id, Address, City, State, ZipCode, Deadline, Weight, Notes, Status, Departure Time, Arrival Time
****************************************************************************************************
24, 5025 State St, Murray, UT, 84107, EOD, 7, , Delivered, 9:05:00, 9:13:00
25,  5383 S 900 East #104, Salt Lake City, UT, 84117, 10:30 AM, 7, DELAYED_905, Delivered, 9:13:00, 9:18:40
22, 6351 South 900 East, Murray, UT, 84121, EOD, 2, , Delivered, 9:18:40, 9:23:00
33, 2530 S 500 E, Salt Lake City, UT, 84106, EOD, 1, , Delivered, 9:23:00, 9:43:00
28, 2835 Main St, Salt Lake City, UT, 84115, EOD, 7, DELAYED_905, Delivered, 9:43:00, 9:46:40
4, 380 W 2880 S, Salt Lake City, UT, 84115, EOD, 4, , Delivered, 9:46:40, 9:50:00
32, 3365 S 900 W, Salt Lake City, UT, 84119, EOD, 1, DELAYED_905, Delivered, 9:50:00, 9:55:40
```

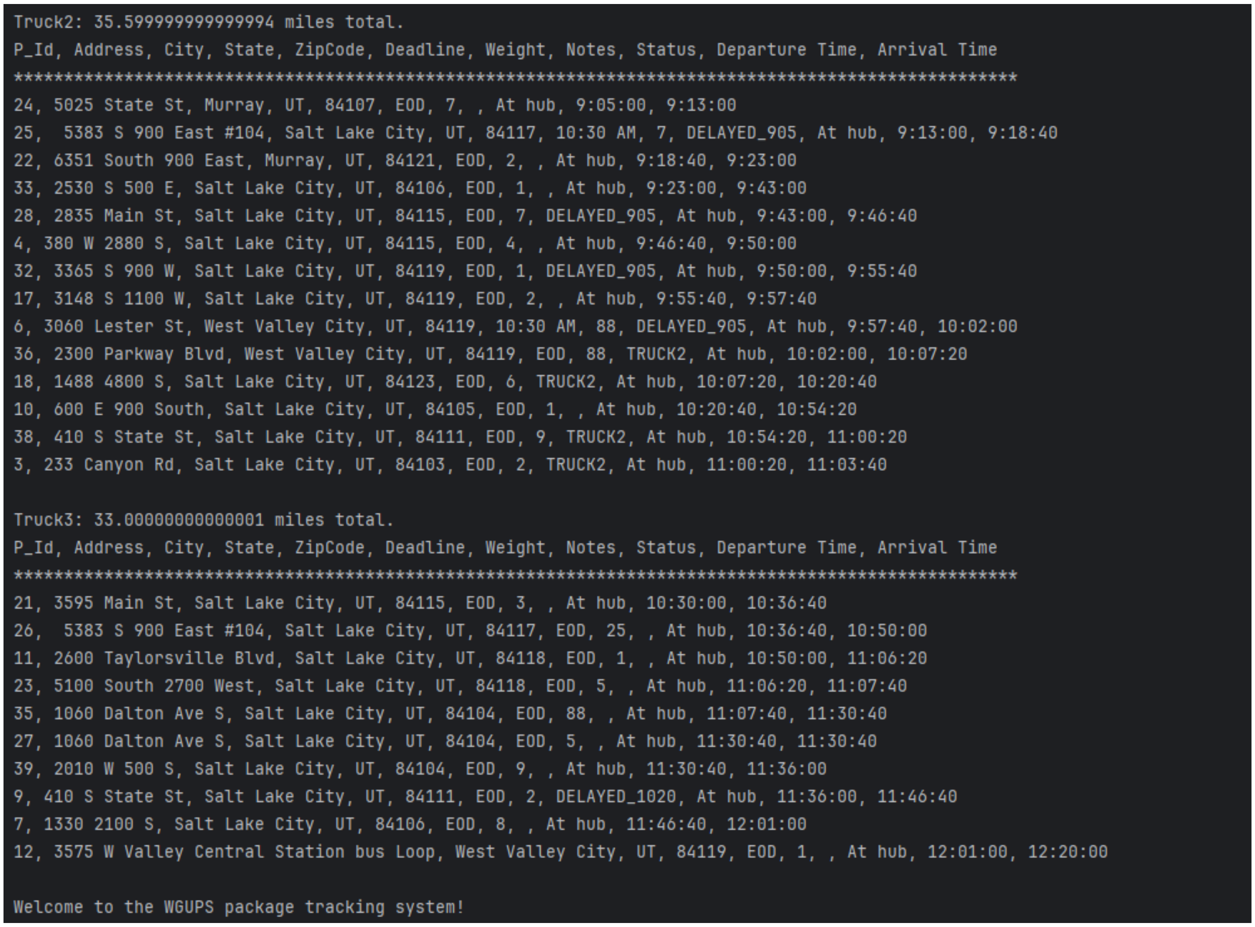


```
Truck2: 35.599999999999994 miles total.
P_Id, Address, City, State, ZipCode, Deadline, Weight, Notes, Status, Departure Time, Arrival Time
****************************************************************************************************
24, 5025 State St, Murray, UT, 84107, EOD, 7, , Delivered, 9:05:00, 9:13:00
25,  5383 S 900 East #104, Salt Lake City, UT, 84117, 10:30 AM, 7, DELAYED_905, Delivered, 9:13:00, 9:18:40
22, 6351 South 900 East, Murray, UT, 84121, EOD, 2, , Delivered, 9:18:40, 9:23:00
33, 2530 S 500 E, Salt Lake City, UT, 84106, EOD, 1, , Delivered, 9:23:00, 9:43:00
28, 2835 Main St, Salt Lake City, UT, 84115, EOD, 7, DELAYED_905, Delivered, 9:43:00, 9:46:40
4, 380 W 2880 S, Salt Lake City, UT, 84115, EOD, 4, , Delivered, 9:46:40, 9:50:00
32, 3365 S 900 W, Salt Lake City, UT, 84119, EOD, 1, DELAYED_905, Delivered, 9:50:00, 9:55:40
17, 3148 S 1100 W, Salt Lake City, UT, 84119, EOD, 2, , Delivered, 9:55:40, 9:57:40
6, 3060 Lester St, West Valley City, UT, 84119, 10:30 AM, 88, DELAYED_905, Delivered, 9:57:40, 10:02:00
36, 2300 Parkway Blvd, West Valley City, UT, 84119, EOD, 88, TRUCK2, Delivered, 10:02:00, 10:07:20
18, 1488 4800 S, Salt Lake City, UT, 84123, EOD, 6, TRUCK2, Delivered, 10:07:20, 10:20:40
10, 600 E 900 South, Salt Lake City, UT, 84105, EOD, 1, , Delivered, 10:20:40, 10:54:20
38, 410 S State St, Salt Lake City, UT, 84111, EOD, 9, TRUCK2, Delivered, 10:54:20, 11:00:20
3, 233 Canyon Rd, Salt Lake City, UT, 84103, EOD, 2, TRUCK2, Delivered, 11:00:20, 11:03:40

Truck3: 33.00000000000001 miles total.
P_Id, Address, City, State, ZipCode, Deadline, Weight, Notes, Status, Departure Time, Arrival Time
****************************************************************************************************
21, 3595 Main St, Salt Lake City, UT, 84115, EOD, 3, , Delivered, 10:30:00, 10:36:40
26,  5383 S 900 East #104, Salt Lake City, UT, 84117, EOD, 25, , Delivered, 10:36:40, 10:50:00
11, 2600 Taylorsville Blvd, Salt Lake City, UT, 84118, EOD, 1, , Delivered, 10:50:00, 11:06:20
23, 5100 South 2700 West, Salt Lake City, UT, 84118, EOD, 5, , Delivered, 11:06:20, 11:07:40
35, 1060 Dalton Ave S, Salt Lake City, UT, 84104, EOD, 88, , Delivered, 11:07:40, 11:30:40
27, 1060 Dalton Ave S, Salt Lake City, UT, 84104, EOD, 5, , Delivered, 11:30:40, 11:30:40
39, 2010 W 500 S, Salt Lake City, UT, 84104, EOD, 9, , Delivered, 11:30:40, 11:36:00
9, 410 S State St, Salt Lake City, UT, 84111, EOD, 2, DELAYED_1020, Delivered, 11:36:00, 11:46:40
7, 1330 2100 S, Salt Lake City, UT, 84106, EOD, 8, , Delivered, 11:46:40, 12:01:00
12, 3575 W Valley Central Station bus Loop, West Valley City, UT, 84119, EOD, 1, , Delivered, 12:01:00, 12:20:00

Welcome to the WGUPS package tracking system!
C950 Assessment - WGUPS
****************************************
1. Print ALL package statuses and total mileage per truck (End of Shift)
2. Print ONE Package Status at a specified time
3. Print ALL package statuses at a specified time
4. Exit the program
****************************************
Please select your desired option:
```

D1. First Status Check

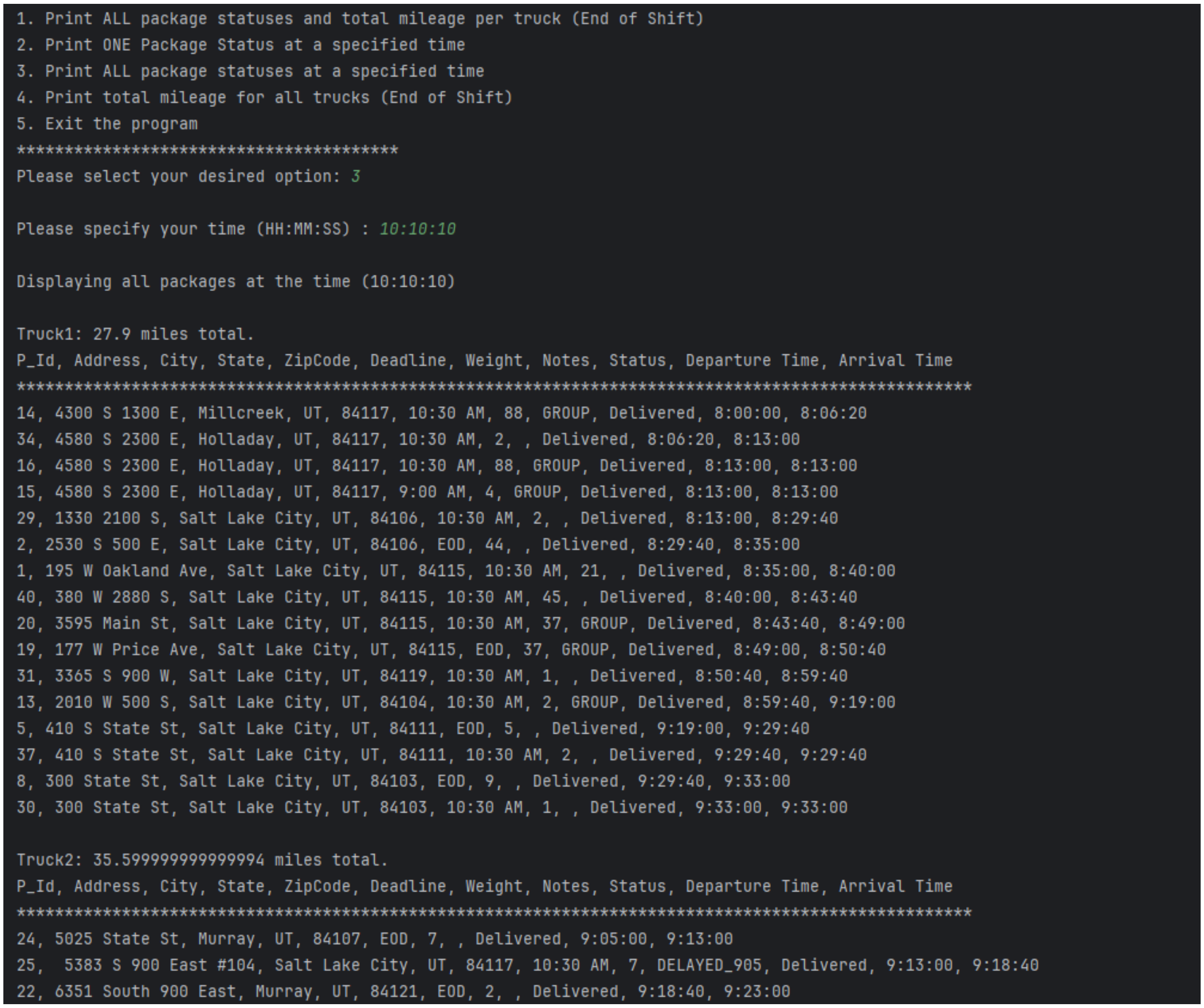


```
Welcome to the WGUPS package tracking system!
C950 Assessment - WGUPS
****************************************
1. Print ALL package statuses and total mileage per truck (End of Shift)
2. Print ONE Package Status at a specified time
3. Print ALL package statuses at a specified time
4. Print total mileage for all trucks (End of Shift)
5. Exit the program
****************************************
Please select your desired option: 3

Please specify your time (HH:MM:SS) : 08:40:00

Displaying all packages at the time (08:40:00)

Truck1: 27.9 miles total.
P_Id, Address, City, State, ZipCode, Deadline, Weight, Notes, Status, Departure Time, Arrival Time
****************************************************************************************************
14, 4300 S 1300 E, Millcreek, UT, 84117, 10:30 AM, 88, GROUP, Delivered, 8:00:00, 8:06:20
34, 4580 S 2300 E, Holladay, UT, 84117, 10:30 AM, 2, , Delivered, 8:06:20, 8:13:00
16, 4580 S 2300 E, Holladay, UT, 84117, 10:30 AM, 88, GROUP, Delivered, 8:13:00, 8:13:00
15, 4580 S 2300 E, Holladay, UT, 84117, 9:00 AM, 4, GROUP, Delivered, 8:13:00, 8:13:00
29, 1330 2100 S, Salt Lake City, UT, 84106, 10:30 AM, 2, , Delivered, 8:13:00, 8:29:40
2, 2530 S 500 E, Salt Lake City, UT, 84106, EOD, 44, , Delivered, 8:29:40, 8:35:00
1, 195 W Oakland Ave, Salt Lake City, UT, 84115, 10:30 AM, 21, , En Route, 8:35:00, 8:40:00
40, 380 W 2880 S, Salt Lake City, UT, 84115, 10:30 AM, 45, , At hub, 8:40:00, 8:43:40
20, 3595 Main St, Salt Lake City, UT, 84115, 10:30 AM, 37, GROUP, At hub, 8:43:40, 8:49:00
19, 177 W Price Ave, Salt Lake City, UT, 84115, EOD, 37, GROUP, At hub, 8:49:00, 8:50:40
31, 3365 S 900 W, Salt Lake City, UT, 84119, 10:30 AM, 1, , At hub, 8:50:40, 8:59:40
13, 2010 W 500 S, Salt Lake City, UT, 84104, 10:30 AM, 2, GROUP, At hub, 8:59:40, 9:19:00
5, 410 S State St, Salt Lake City, UT, 84111, EOD, 5, , At hub, 9:19:00, 9:29:40
37, 410 S State St, Salt Lake City, UT, 84111, 10:30 AM, 2, , At hub, 9:29:40, 9:29:40
8, 300 State St, Salt Lake City, UT, 84103, EOD, 9, , At hub, 9:29:40, 9:33:00
30, 300 State St, Salt Lake City, UT, 84103, 10:30 AM, 1, , At hub, 9:33:00, 9:33:00

Truck2: 35.599999999999994 miles total.
P_Id, Address, City, State, ZipCode, Deadline, Weight, Notes, Status, Departure Time, Arrival Time
****************************************************************************************************
```

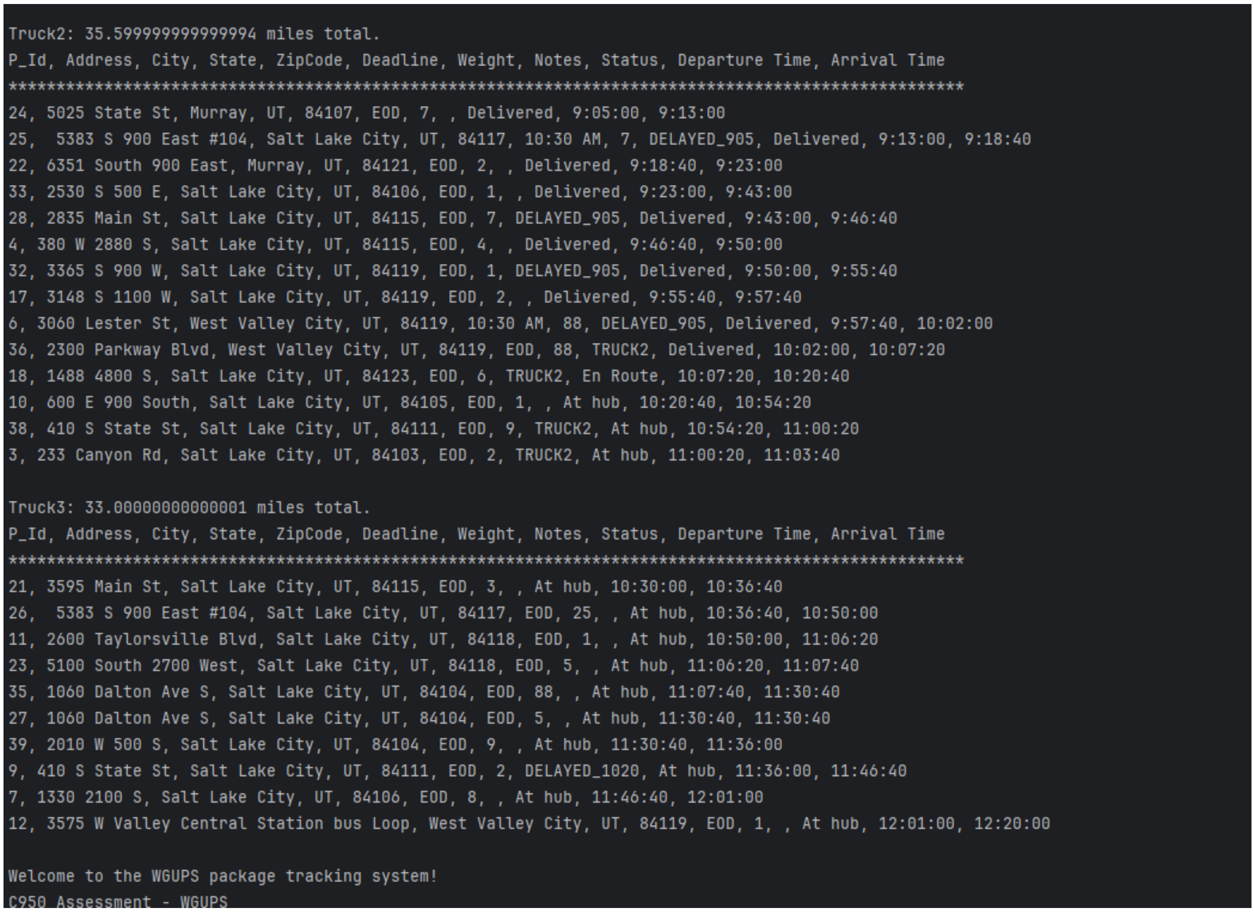


```
Truck2: 35.599999999999994 miles total.
P_Id, Address, City, State, ZipCode, Deadline, Weight, Notes, Status, Departure Time, Arrival Time
****************************************************************************************************
24, 5025 State St, Murray, UT, 84107, EOD, 7, , At hub, 9:05:00, 9:13:00
25,  5383 S 900 East #104, Salt Lake City, UT, 84117, 10:30 AM, 7, DELAYED_905, At hub, 9:13:00, 9:18:40
22, 6351 South 900 East, Murray, UT, 84121, EOD, 2, , At hub, 9:18:40, 9:23:00
33, 2530 S 500 E, Salt Lake City, UT, 84106, EOD, 1, , At hub, 9:23:00, 9:43:00
28, 2835 Main St, Salt Lake City, UT, 84115, EOD, 7, DELAYED_905, At hub, 9:43:00, 9:46:40
4, 380 W 2880 S, Salt Lake City, UT, 84115, EOD, 4, , At hub, 9:46:40, 9:50:00
32, 3365 S 900 W, Salt Lake City, UT, 84119, EOD, 1, DELAYED_905, At hub, 9:50:00, 9:55:40
17, 3148 S 1100 W, Salt Lake City, UT, 84119, EOD, 2, , At hub, 9:55:40, 9:57:40
6, 3060 Lester St, West Valley City, UT, 84119, 10:30 AM, 88, DELAYED_905, At hub, 9:57:40, 10:02:00
36, 2300 Parkway Blvd, West Valley City, UT, 84119, EOD, 88, TRUCK2, At hub, 10:02:00, 10:07:20
18, 1488 4800 S, Salt Lake City, UT, 84123, EOD, 6, TRUCK2, At hub, 10:07:20, 10:20:40
10, 600 E 900 South, Salt Lake City, UT, 84105, EOD, 1, , At hub, 10:20:40, 10:54:20
38, 410 S State St, Salt Lake City, UT, 84111, EOD, 9, TRUCK2, At hub, 10:54:20, 11:00:20
3, 233 Canyon Rd, Salt Lake City, UT, 84103, EOD, 2, TRUCK2, At hub, 11:00:20, 11:03:40

Truck3: 33.00000000000001 miles total.
P_Id, Address, City, State, ZipCode, Deadline, Weight, Notes, Status, Departure Time, Arrival Time
****************************************************************************************************
21, 3595 Main St, Salt Lake City, UT, 84115, EOD, 3, , At hub, 10:30:00, 10:36:40
26,  5383 S 900 East #104, Salt Lake City, UT, 84117, EOD, 25, , At hub, 10:36:40, 10:50:00
11, 2600 Taylorsville Blvd, Salt Lake City, UT, 84118, EOD, 1, , At hub, 10:50:00, 11:06:20
23, 5100 South 2700 West, Salt Lake City, UT, 84118, EOD, 5, , At hub, 11:06:20, 11:07:40
35, 1060 Dalton Ave S, Salt Lake City, UT, 84104, EOD, 88, , At hub, 11:07:40, 11:30:40
27, 1060 Dalton Ave S, Salt Lake City, UT, 84104, EOD, 5, , At hub, 11:30:40, 11:30:40
39, 2010 W 500 S, Salt Lake City, UT, 84104, EOD, 9, , At hub, 11:30:40, 11:36:00
9, 410 S State St, Salt Lake City, UT, 84111, EOD, 2, DELAYED_1020, At hub, 11:36:00, 11:46:40
7, 1330 2100 S, Salt Lake City, UT, 84106, EOD, 8, , At hub, 11:46:40, 12:01:00
12, 3575 W Valley Central Station bus Loop, West Valley City, UT, 84119, EOD, 1, , At hub, 12:01:00, 12:20:00

Welcome to the WGUPS package tracking system!
```

D2. Second Status Check

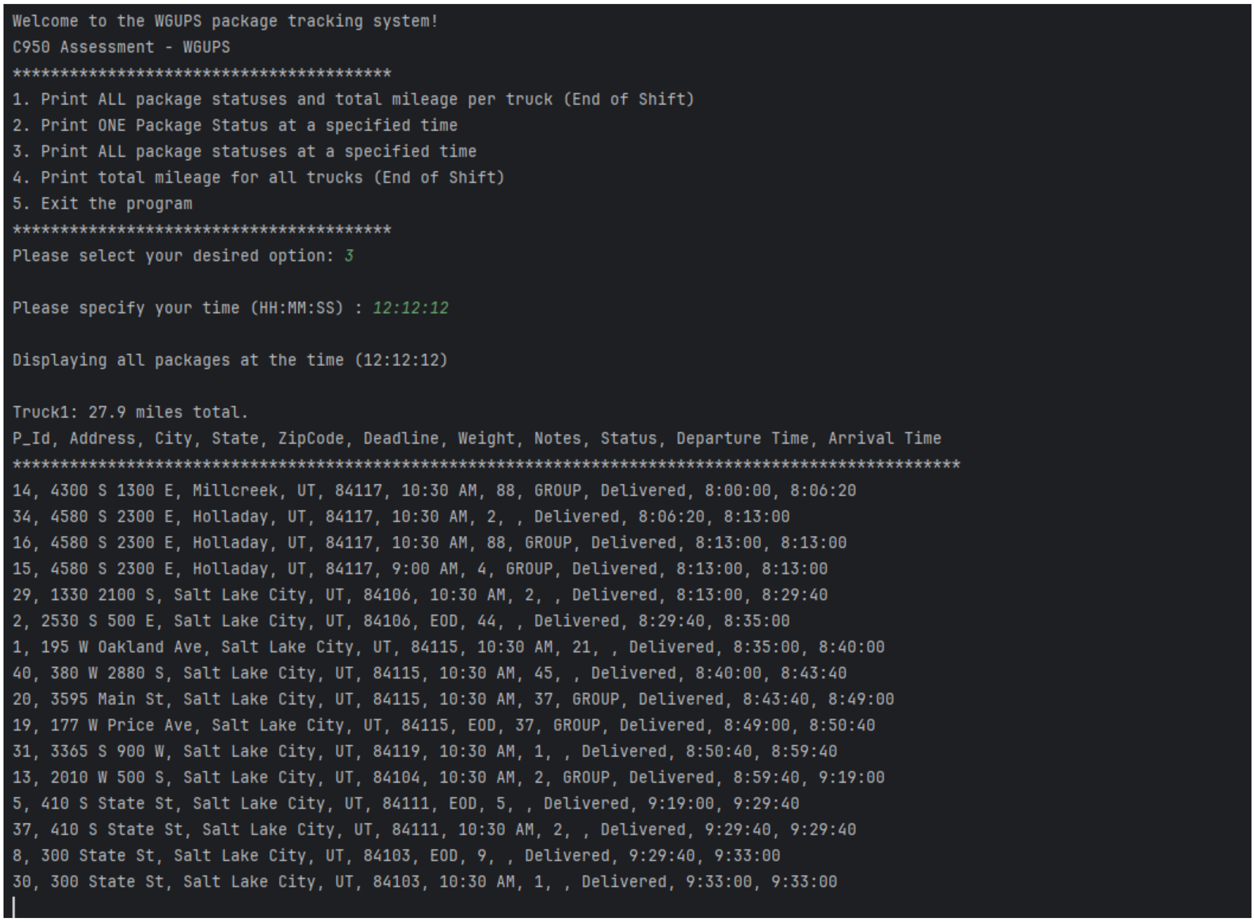


```
1. Print ALL package statuses and total mileage per truck (End of Shift)
2. Print ONE Package Status at a specified time
3. Print ALL package statuses at a specified time
4. Print total mileage for all trucks (End of Shift)
5. Exit the program
****************************************
Please select your desired option: 3

Please specify your time (HH:MM:SS) : 10:10:10

Displaying all packages at the time (10:10:10)

Truck1: 27.9 miles total.
P_Id, Address, City, State, ZipCode, Deadline, Weight, Notes, Status, Departure Time, Arrival Time
****************************************************************************************************
14, 4300 S 1300 E, Millcreek, UT, 84117, 10:30 AM, 88, GROUP, Delivered, 8:00:00, 8:06:20
34, 4580 S 2300 E, Holladay, UT, 84117, 10:30 AM, 2, , Delivered, 8:06:20, 8:13:00
16, 4580 S 2300 E, Holladay, UT, 84117, 10:30 AM, 88, GROUP, Delivered, 8:13:00, 8:13:00
15, 4580 S 2300 E, Holladay, UT, 84117, 9:00 AM, 4, GROUP, Delivered, 8:13:00, 8:13:00
29, 1330 2100 S, Salt Lake City, UT, 84106, 10:30 AM, 2, , Delivered, 8:13:00, 8:29:40
2, 2530 S 500 E, Salt Lake City, UT, 84106, EOD, 44, , Delivered, 8:29:40, 8:35:00
1, 195 W Oakland Ave, Salt Lake City, UT, 84115, 10:30 AM, 21, , Delivered, 8:35:00, 8:40:00
40, 380 W 2880 S, Salt Lake City, UT, 84115, 10:30 AM, 45, , Delivered, 8:40:00, 8:43:40
20, 3595 Main St, Salt Lake City, UT, 84115, 10:30 AM, 37, GROUP, Delivered, 8:43:40, 8:49:00
19, 177 W Price Ave, Salt Lake City, UT, 84115, EOD, 37, GROUP, Delivered, 8:49:00, 8:50:40
31, 3365 S 900 W, Salt Lake City, UT, 84119, 10:30 AM, 1, , Delivered, 8:50:40, 8:59:40
13, 2010 W 500 S, Salt Lake City, UT, 84104, 10:30 AM, 2, GROUP, Delivered, 8:59:40, 9:19:00
5, 410 S State St, Salt Lake City, UT, 84111, EOD, 5, , Delivered, 9:19:00, 9:29:40
37, 410 S State St, Salt Lake City, UT, 84111, 10:30 AM, 2, , Delivered, 9:29:40, 9:29:40
8, 300 State St, Salt Lake City, UT, 84103, EOD, 9, , Delivered, 9:29:40, 9:33:00
30, 300 State St, Salt Lake City, UT, 84103, 10:30 AM, 1, , Delivered, 9:33:00, 9:33:00

Truck2: 35.599999999999994 miles total.
P_Id, Address, City, State, ZipCode, Deadline, Weight, Notes, Status, Departure Time, Arrival Time
****************************************************************************************************
24, 5025 State St, Murray, UT, 84107, EOD, 7, , Delivered, 9:05:00, 9:13:00
25,  5383 S 900 East #104, Salt Lake City, UT, 84117, 10:30 AM, 7, DELAYED_905, Delivered, 9:13:00, 9:18:40
22, 6351 South 900 East, Murray, UT, 84121, EOD, 2, , Delivered, 9:18:40, 9:23:00
```

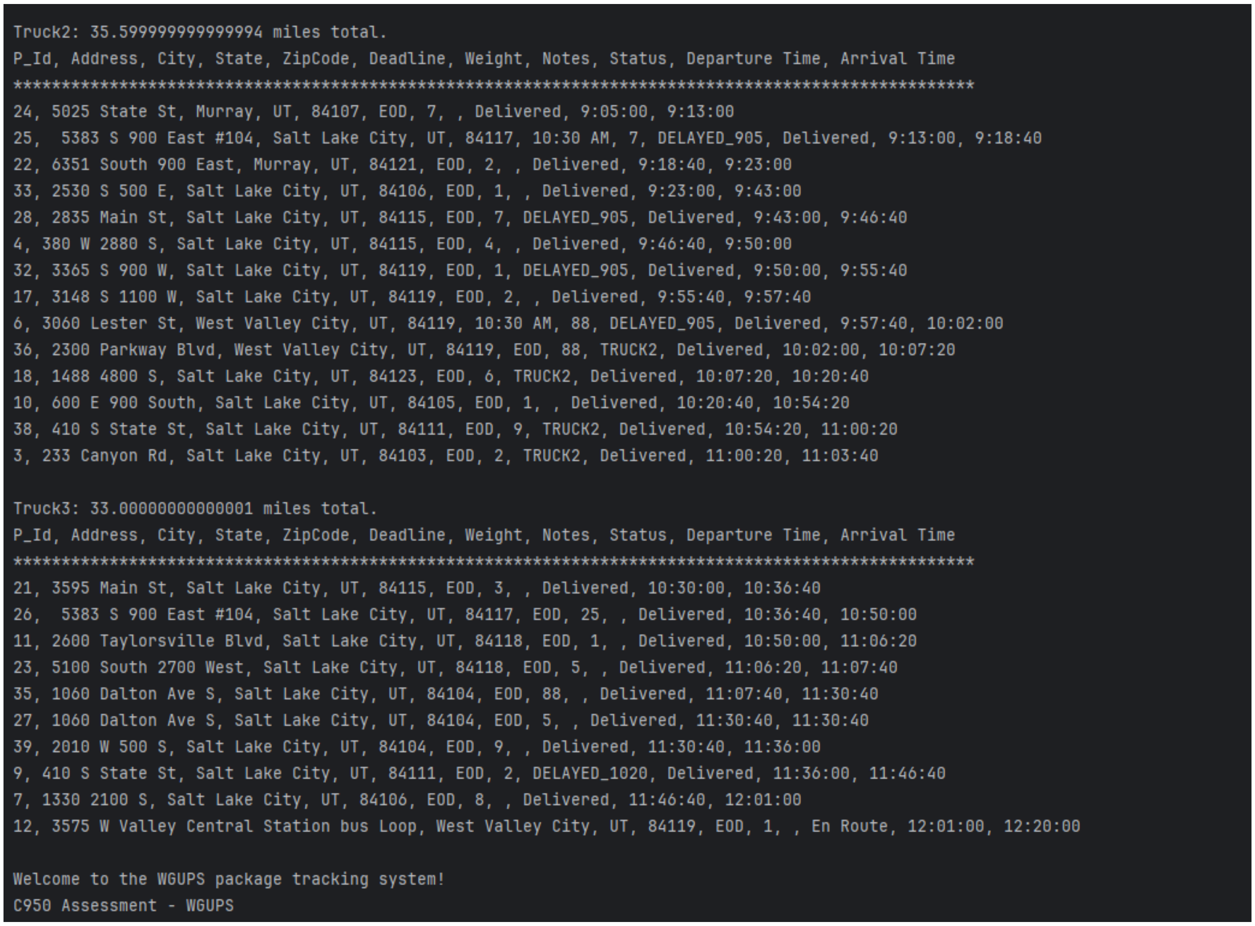


```
Truck2: 35.599999999999994 miles total.
P_Id, Address, City, State, ZipCode, Deadline, Weight, Notes, Status, Departure Time, Arrival Time
****************************************************************************************************
24, 5025 State St, Murray, UT, 84107, EOD, 7, , Delivered, 9:05:00, 9:13:00
25,  5383 S 900 East #104, Salt Lake City, UT, 84117, 10:30 AM, 7, DELAYED_905, Delivered, 9:13:00, 9:18:40
22, 6351 South 900 East, Murray, UT, 84121, EOD, 2, , Delivered, 9:18:40, 9:23:00
33, 2530 S 500 E, Salt Lake City, UT, 84106, EOD, 1, , Delivered, 9:23:00, 9:43:00
28, 2835 Main St, Salt Lake City, UT, 84115, EOD, 7, DELAYED_905, Delivered, 9:43:00, 9:46:40
4, 380 W 2880 S, Salt Lake City, UT, 84115, EOD, 4, , Delivered, 9:46:40, 9:50:00
32, 3365 S 900 W, Salt Lake City, UT, 84119, EOD, 1, DELAYED_905, Delivered, 9:50:00, 9:55:40
17, 3148 S 1100 W, Salt Lake City, UT, 84119, EOD, 2, , Delivered, 9:55:40, 9:57:40
6, 3060 Lester St, West Valley City, UT, 84119, 10:30 AM, 88, DELAYED_905, Delivered, 9:57:40, 10:02:00
36, 2300 Parkway Blvd, West Valley City, UT, 84119, EOD, 88, TRUCK2, Delivered, 10:02:00, 10:07:20
18, 1488 4800 S, Salt Lake City, UT, 84123, EOD, 6, TRUCK2, En Route, 10:07:20, 10:20:40
10, 600 E 900 South, Salt Lake City, UT, 84105, EOD, 1, , At hub, 10:20:40, 10:54:20
38, 410 S State St, Salt Lake City, UT, 84111, EOD, 9, TRUCK2, At hub, 10:54:20, 11:00:20
3, 233 Canyon Rd, Salt Lake City, UT, 84103, EOD, 2, TRUCK2, At hub, 11:00:20, 11:03:40

Truck3: 33.00000000000001 miles total.
P_Id, Address, City, State, ZipCode, Deadline, Weight, Notes, Status, Departure Time, Arrival Time
****************************************************************************************************
21, 3595 Main St, Salt Lake City, UT, 84115, EOD, 3, , At hub, 10:30:00, 10:36:40
26,  5383 S 900 East #104, Salt Lake City, UT, 84117, EOD, 25, , At hub, 10:36:40, 10:50:00
11, 2600 Taylorsville Blvd, Salt Lake City, UT, 84118, EOD, 1, , At hub, 10:50:00, 11:06:20
23, 5100 South 2700 West, Salt Lake City, UT, 84118, EOD, 5, , At hub, 11:06:20, 11:07:40
35, 1060 Dalton Ave S, Salt Lake City, UT, 84104, EOD, 88, , At hub, 11:07:40, 11:30:40
27, 1060 Dalton Ave S, Salt Lake City, UT, 84104, EOD, 5, , At hub, 11:30:40, 11:30:40
39, 2010 W 500 S, Salt Lake City, UT, 84104, EOD, 9, , At hub, 11:30:40, 11:36:00
9, 410 S State St, Salt Lake City, UT, 84111, EOD, 2, DELAYED_1020, At hub, 11:36:00, 11:46:40
7, 1330 2100 S, Salt Lake City, UT, 84106, EOD, 8, , At hub, 11:46:40, 12:01:00
12, 3575 W Valley Central Station bus Loop, West Valley City, UT, 84119, EOD, 1, , At hub, 12:01:00, 12:20:00

Welcome to the WGUPS package tracking system!
C950 Assessment - WGUPS
```

D3. Third Status Check

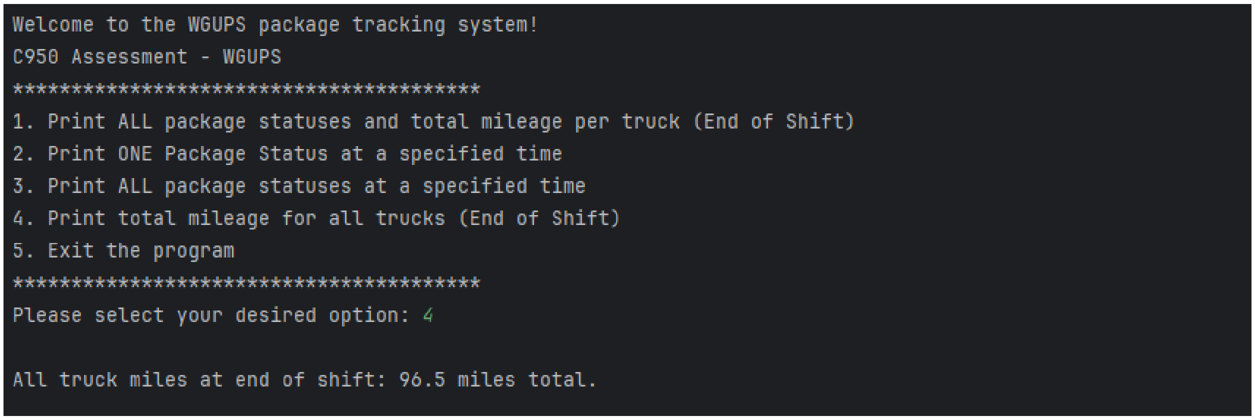


```
Welcome to the WGUPS package tracking system!
C950 Assessment - WGUPS
****************************************
1. Print ALL package statuses and total mileage per truck (End of Shift)
2. Print ONE Package Status at a specified time
3. Print ALL package statuses at a specified time
4. Print total mileage for all trucks (End of Shift)
5. Exit the program
****************************************
Please select your desired option: 3

Please specify your time (HH:MM:SS) : 12:12:12

Displaying all packages at the time (12:12:12)

Truck1: 27.9 miles total.
P_Id, Address, City, State, ZipCode, Deadline, Weight, Notes, Status, Departure Time, Arrival Time
****************************************************************************************************
14, 4300 S 1300 E, Millcreek, UT, 84117, 10:30 AM, 88, GROUP, Delivered, 8:00:00, 8:06:20
34, 4580 S 2300 E, Holladay, UT, 84117, 10:30 AM, 2, , Delivered, 8:06:20, 8:13:00
16, 4580 S 2300 E, Holladay, UT, 84117, 10:30 AM, 88, GROUP, Delivered, 8:13:00, 8:13:00
15, 4580 S 2300 E, Holladay, UT, 84117, 9:00 AM, 4, GROUP, Delivered, 8:13:00, 8:13:00
29, 1330 2100 S, Salt Lake City, UT, 84106, 10:30 AM, 2, , Delivered, 8:13:00, 8:29:40
2, 2530 S 500 E, Salt Lake City, UT, 84106, EOD, 44, , Delivered, 8:29:40, 8:35:00
1, 195 W Oakland Ave, Salt Lake City, UT, 84115, 10:30 AM, 21, , Delivered, 8:35:00, 8:40:00
40, 380 W 2880 S, Salt Lake City, UT, 84115, 10:30 AM, 45, , Delivered, 8:40:00, 8:43:40
20, 3595 Main St, Salt Lake City, UT, 84115, 10:30 AM, 37, GROUP, Delivered, 8:43:40, 8:49:00
19, 177 W Price Ave, Salt Lake City, UT, 84115, EOD, 37, GROUP, Delivered, 8:49:00, 8:50:40
31, 3365 S 900 W, Salt Lake City, UT, 84119, 10:30 AM, 1, , Delivered, 8:50:40, 8:59:40
13, 2010 W 500 S, Salt Lake City, UT, 84104, 10:30 AM, 2, GROUP, Delivered, 8:59:40, 9:19:00
5, 410 S State St, Salt Lake City, UT, 84111, EOD, 5, , Delivered, 9:19:00, 9:29:40
37, 410 S State St, Salt Lake City, UT, 84111, 10:30 AM, 2, , Delivered, 9:29:40, 9:29:40
8, 300 State St, Salt Lake City, UT, 84103, EOD, 9, , Delivered, 9:29:40, 9:33:00
30, 300 State St, Salt Lake City, UT, 84103, 10:30 AM, 1, , Delivered, 9:33:00, 9:33:00
```

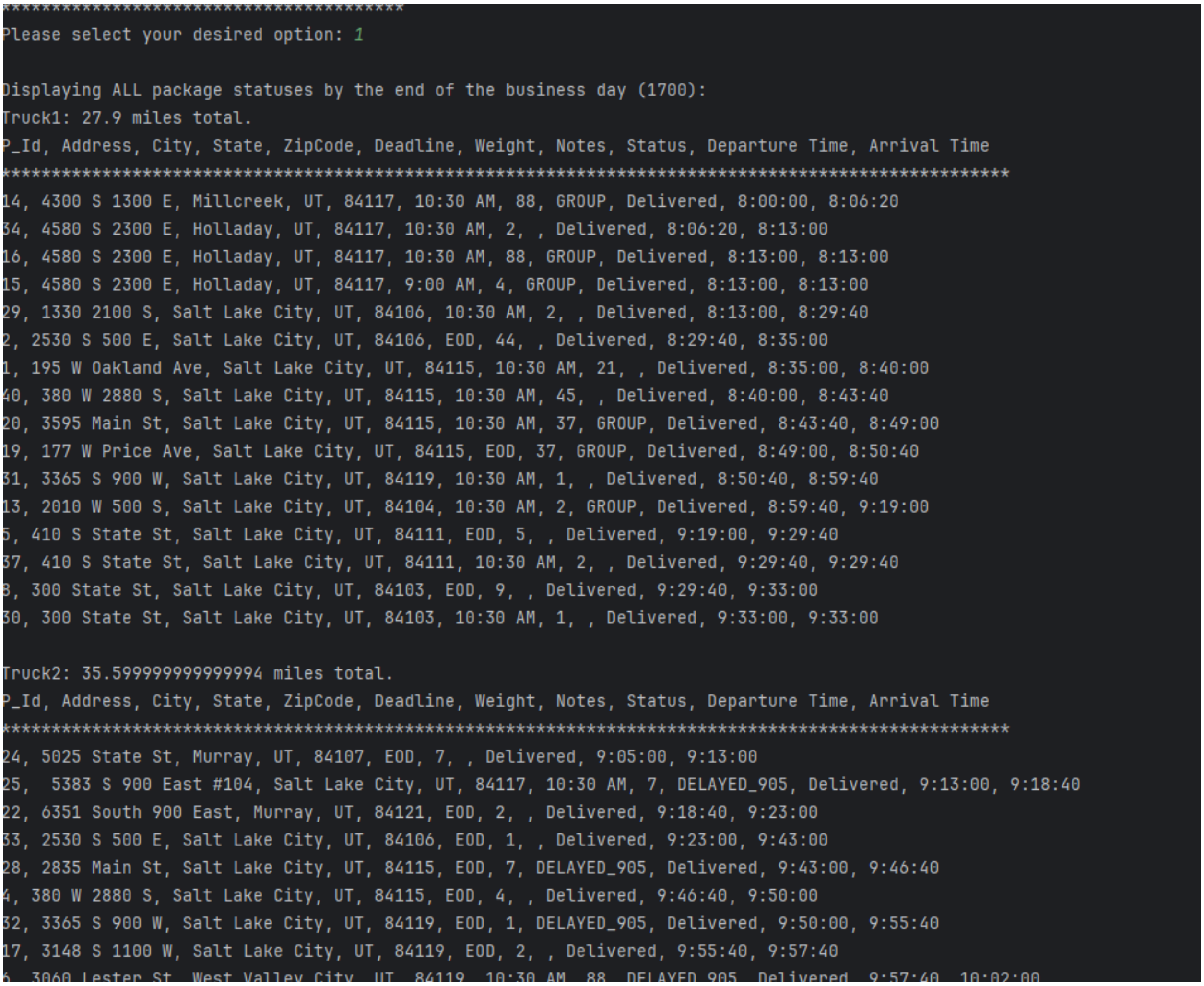


```
Truck2: 35.599999999999994 miles total.
P_Id, Address, City, State, ZipCode, Deadline, Weight, Notes, Status, Departure Time, Arrival Time
***************************************************************************************************
24, 5025 State St, Murray, UT, 84107, EOD, 7, , Delivered, 9:05:00, 9:13:00
25,  5383 S 900 East #104, Salt Lake City, UT, 84117, 10:30 AM, 7, DELAYED_905, Delivered, 9:13:00, 9:18:40
22, 6351 South 900 East, Murray, UT, 84121, EOD, 2, , Delivered, 9:18:40, 9:23:00
33, 2530 S 500 E, Salt Lake City, UT, 84106, EOD, 1, , Delivered, 9:23:00, 9:43:00
28, 2835 Main St, Salt Lake City, UT, 84115, EOD, 7, DELAYED_905, Delivered, 9:43:00, 9:46:40
4, 380 W 2880 S, Salt Lake City, UT, 84115, EOD, 4, , Delivered, 9:46:40, 9:50:00
32, 3365 S 900 W, Salt Lake City, UT, 84119, EOD, 1, DELAYED_905, Delivered, 9:50:00, 9:55:40
17, 3148 S 1100 W, Salt Lake City, UT, 84119, EOD, 2, , Delivered, 9:55:40, 9:57:40
6, 3060 Lester St, West Valley City, UT, 84119, 10:30 AM, 88, DELAYED_905, Delivered, 9:57:40, 10:02:00
36, 2300 Parkway Blvd, West Valley City, UT, 84119, EOD, 88, TRUCK2, Delivered, 10:02:00, 10:07:20
18, 1488 4800 S, Salt Lake City, UT, 84123, EOD, 6, TRUCK2, Delivered, 10:07:20, 10:20:40
10, 600 E 900 South, Salt Lake City, UT, 84105, EOD, 1, , Delivered, 10:20:40, 10:54:20
38, 410 S State St, Salt Lake City, UT, 84111, EOD, 9, TRUCK2, Delivered, 10:54:20, 11:00:20
3, 233 Canyon Rd, Salt Lake City, UT, 84103, EOD, 2, TRUCK2, Delivered, 11:00:20, 11:03:40

Truck3: 33.00000000000001 miles total.
P_Id, Address, City, State, ZipCode, Deadline, Weight, Notes, Status, Departure Time, Arrival Time
***************************************************************************************************
21, 3595 Main St, Salt Lake City, UT, 84115, EOD, 3, , Delivered, 10:30:00, 10:36:40
26,  5383 S 900 East #104, Salt Lake City, UT, 84117, EOD, 25, , Delivered, 10:36:40, 10:50:00
11, 2600 Taylorsville Blvd, Salt Lake City, UT, 84118, EOD, 1, , Delivered, 10:50:00, 11:06:20
23, 5100 South 2700 West, Salt Lake City, UT, 84118, EOD, 5, , Delivered, 11:06:20, 11:07:40
35, 1060 Dalton Ave S, Salt Lake City, UT, 84104, EOD, 88, , Delivered, 11:07:40, 11:30:40
27, 1060 Dalton Ave S, Salt Lake City, UT, 84104, EOD, 5, , Delivered, 11:30:40, 11:30:40
39, 2010 W 500 S, Salt Lake City, UT, 84104, EOD, 9, , Delivered, 11:30:40, 11:36:00
9, 410 S State St, Salt Lake City, UT, 84111, EOD, 2, DELAYED_1020, Delivered, 11:36:00, 11:46:40
7, 1330 2100 S, Salt Lake City, UT, 84106, EOD, 8, , Delivered, 11:46:40, 12:01:00
12, 3575 W Valley Central Station bus Loop, West Valley City, UT, 84119, EOD, 1, , En Route, 12:01:00, 12:20:00

Welcome to the WGUPS package tracking system!
C950 Assessment - WGUPS
```

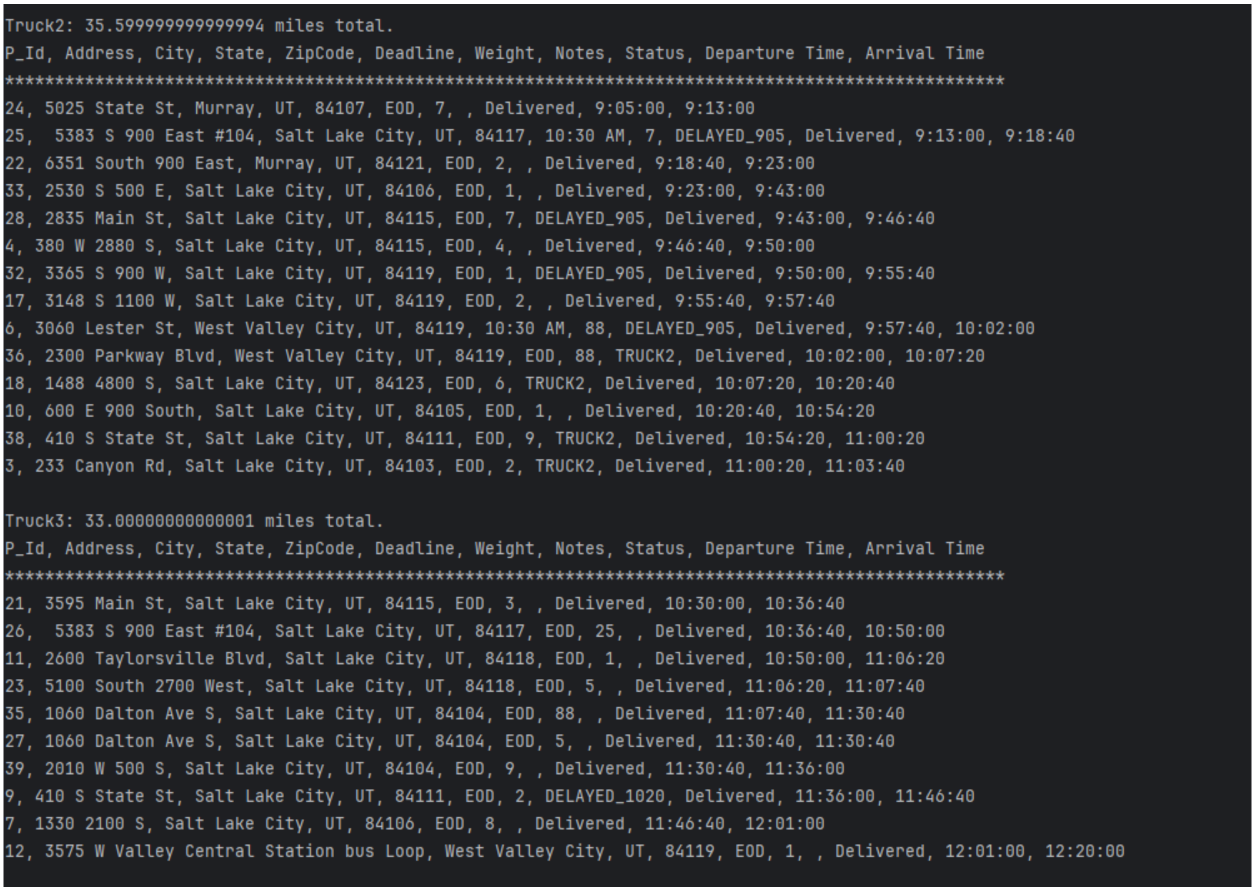
E. Screenshot of Code Execution



```
Welcome to the WGUPS package tracking system!
C950 Assessment - WGUPS
****************************************
1. Print ALL package statuses and total mileage per truck (End of Shift)
2. Print ONE Package Status at a specified time
3. Print ALL package statuses at a specified time
4. Print total mileage for all trucks (End of Shift)
5. Exit the program
****************************************
Please select your desired option: 4

All truck miles at end of shift: 96.5 miles total.
```



```
****************************************
Please select your desired option: 1

Displaying ALL package statuses by the end of the business day (1700):
Truck1: 27.9 miles total.
P_Id, Address, City, State, ZipCode, Deadline, Weight, Notes, Status, Departure Time, Arrival Time
****************************************************************************************************
14, 4300 S 1300 E, Millcreek, UT, 84117, 10:30 AM, 88, GROUP, Delivered, 8:00:00, 8:06:20
34, 4580 S 2300 E, Holladay, UT, 84117, 10:30 AM, 2, , Delivered, 8:06:20, 8:13:00
16, 4580 S 2300 E, Holladay, UT, 84117, 10:30 AM, 88, GROUP, Delivered, 8:13:00, 8:13:00
15, 4580 S 2300 E, Holladay, UT, 84117, 9:00 AM, 4, GROUP, Delivered, 8:13:00, 8:13:00
29, 1330 2100 S, Salt Lake City, UT, 84106, 10:30 AM, 2, , Delivered, 8:13:00, 8:29:40
2, 2530 S 500 E, Salt Lake City, UT, 84106, EOD, 44, , Delivered, 8:29:40, 8:35:00
1, 195 W Oakland Ave, Salt Lake City, UT, 84115, 10:30 AM, 21, , Delivered, 8:35:00, 8:40:00
40, 380 W 2880 S, Salt Lake City, UT, 84115, 10:30 AM, 45, , Delivered, 8:40:00, 8:43:40
20, 3595 Main St, Salt Lake City, UT, 84115, 10:30 AM, 37, GROUP, Delivered, 8:43:40, 8:49:00
19, 177 W Price Ave, Salt Lake City, UT, 84115, EOD, 37, GROUP, Delivered, 8:49:00, 8:50:40
31, 3365 S 900 W, Salt Lake City, UT, 84119, 10:30 AM, 1, , Delivered, 8:50:40, 8:59:40
13, 2010 W 500 S, Salt Lake City, UT, 84104, 10:30 AM, 2, GROUP, Delivered, 8:59:40, 9:19:00
5, 410 S State St, Salt Lake City, UT, 84111, EOD, 5, , Delivered, 9:19:00, 9:29:40
37, 410 S State St, Salt Lake City, UT, 84111, 10:30 AM, 2, , Delivered, 9:29:40, 9:29:40
8, 300 State St, Salt Lake City, UT, 84103, EOD, 9, , Delivered, 9:29:40, 9:33:00
30, 300 State St, Salt Lake City, UT, 84103, 10:30 AM, 1, , Delivered, 9:33:00, 9:33:00

Truck2: 35.599999999999994 miles total.
P_Id, Address, City, State, ZipCode, Deadline, Weight, Notes, Status, Departure Time, Arrival Time
****************************************************************************************************
24, 5025 State St, Murray, UT, 84107, EOD, 7, , Delivered, 9:05:00, 9:13:00
25,  5383 S 900 East #104, Salt Lake City, UT, 84117, 10:30 AM, 7, DELAYED_905, Delivered, 9:13:00, 9:18:40
22, 6351 South 900 East, Murray, UT, 84121, EOD, 2, , Delivered, 9:18:40, 9:23:00
33, 2530 S 500 E, Salt Lake City, UT, 84106, EOD, 1, , Delivered, 9:23:00, 9:43:00
28, 2835 Main St, Salt Lake City, UT, 84115, EOD, 7, DELAYED_905, Delivered, 9:43:00, 9:46:40
4, 380 W 2880 S, Salt Lake City, UT, 84115, EOD, 4, , Delivered, 9:46:40, 9:50:00
32, 3365 S 900 W, Salt Lake City, UT, 84119, EOD, 1, DELAYED_905, Delivered, 9:50:00, 9:55:40
17, 3148 S 1100 W, Salt Lake City, UT, 84119, EOD, 2, , Delivered, 9:55:40, 9:57:40
6, 3060 Lester St, West Valley City, UT, 84119, 10:30 AM, 88, DELAYED_905, Delivered, 9:57:40, 10:02:00
```



```
Truck2: 35.599999999999994 miles total.
P_Id, Address, City, State, ZipCode, Deadline, Weight, Notes, Status, Departure Time, Arrival Time
****************************************************************************************************
24, 5025 State St, Murray, UT, 84107, EOD, 7, , Delivered, 9:05:00, 9:13:00
25,  5383 S 900 East #104, Salt Lake City, UT, 84117, 10:30 AM, 7, DELAYED_905, Delivered, 9:13:00, 9:18:40
22, 6351 South 900 East, Murray, UT, 84121, EOD, 2, , Delivered, 9:18:40, 9:23:00
33, 2530 S 500 E, Salt Lake City, UT, 84106, EOD, 1, , Delivered, 9:23:00, 9:43:00
28, 2835 Main St, Salt Lake City, UT, 84115, EOD, 7, DELAYED_905, Delivered, 9:43:00, 9:46:40
4, 380 W 2880 S, Salt Lake City, UT, 84115, EOD, 4, , Delivered, 9:46:40, 9:50:00
32, 3365 S 900 W, Salt Lake City, UT, 84119, EOD, 1, DELAYED_905, Delivered, 9:50:00, 9:55:40
17, 3148 S 1100 W, Salt Lake City, UT, 84119, EOD, 2, , Delivered, 9:55:40, 9:57:40
6, 3060 Lester St, West Valley City, UT, 84119, 10:30 AM, 88, DELAYED_905, Delivered, 9:57:40, 10:02:00
36, 2300 Parkway Blvd, West Valley City, UT, 84119, EOD, 88, TRUCK2, Delivered, 10:02:00, 10:07:20
18, 1488 4800 S, Salt Lake City, UT, 84123, EOD, 6, TRUCK2, Delivered, 10:07:20, 10:20:40
10, 600 E 900 South, Salt Lake City, UT, 84105, EOD, 1, , Delivered, 10:20:40, 10:54:20
38, 410 S State St, Salt Lake City, UT, 84111, EOD, 9, TRUCK2, Delivered, 10:54:20, 11:00:20
3, 233 Canyon Rd, Salt Lake City, UT, 84103, EOD, 2, TRUCK2, Delivered, 11:00:20, 11:03:40

Truck3: 33.00000000000001 miles total.
P_Id, Address, City, State, ZipCode, Deadline, Weight, Notes, Status, Departure Time, Arrival Time
****************************************************************************************************
21, 3595 Main St, Salt Lake City, UT, 84115, EOD, 3, , Delivered, 10:30:00, 10:36:40
26,  5383 S 900 East #104, Salt Lake City, UT, 84117, EOD, 25, , Delivered, 10:36:40, 10:50:00
11, 2600 Taylorsville Blvd, Salt Lake City, UT, 84118, EOD, 1, , Delivered, 10:50:00, 11:06:20
23, 5100 South 2700 West, Salt Lake City, UT, 84118, EOD, 5, , Delivered, 11:06:20, 11:07:40
35, 1060 Dalton Ave S, Salt Lake City, UT, 84104, EOD, 88, , Delivered, 11:07:40, 11:30:40
27, 1060 Dalton Ave S, Salt Lake City, UT, 84104, EOD, 5, , Delivered, 11:30:40, 11:30:40
39, 2010 W 500 S, Salt Lake City, UT, 84104, EOD, 9, , Delivered, 11:30:40, 11:36:00
9, 410 S State St, Salt Lake City, UT, 84111, EOD, 2, DELAYED_1020, Delivered, 11:36:00, 11:46:40
7, 1330 2100 S, Salt Lake City, UT, 84106, EOD, 8, , Delivered, 11:46:40, 12:01:00
12, 3575 W Valley Central Station bus Loop, West Valley City, UT, 84119, EOD, 1, , Delivered, 12:01:00, 12:20:00
```

F1. Strengths of the Chosen Algorithm

The nearest neighbor (NN) algorithm operates on a straightforward principle: at each step, it selects the closest available option based on a predefined distance metric. This simplicity makes it easy to understand, implement, and debug. Its lack of computational complexity ensures it can be deployed quickly in real-world scenarios, even by developers with limited experience.

The NN algorithm excels in finding locally optimal solutions rapidly. For tasks where quick decision-making is critical, such as routing or classification, this approach can deliver satisfactory results without requiring extensive computational resources. Its greedy nature ensures progress at each step, particularly useful in scenarios with tight time constraints.

F2. Verification of Algorithm

The NN algorithm uses effective routing, accomplishes time-sensitive deliveries, and adaptability. The algorithm effectively minimizes the travel distance by always selecting the nearest unvisited delivery location, keeping the total distance under the 140-mile constraint. By prioritizing nearby deliveries, the NN algorithm supports meeting package deadlines efficiently, which is crucial for WGUPS's commitment to timely delivery. The NN algorithm's flexibility allows it to handle real-time updates, such as the corrected address for Package #9 at 10:20 a.m., ensuring the routing plan remains valid.

F3. Other Possible Algorithms

Two other algorithms are the A* algorithm and simulated Annealing.

F3a. Algorithm Differences

A* Algorithm (greedy)- A* is a pathfinding and graph traversal algorithm that combines the benefits of Dijkstra's algorithm and a heuristic approach. It uses a cost function that incorporates both the distance traveled and an estimate of the remaining distance to the goal. While the NN algorithm relies purely on local decisions, A* considers the global perspective by incorporating heuristics, making it more effective for complex routing problems with multiple constraints.

Simulated Annealing (metaheuristic)- It is a probabilistic optimization algorithm inspired by the annealing process in metallurgy. Baeldung (baeldung) states, “SA iteratively improves the current solution by randomly perturbing it and accepting the perturbation with a certain probability.” It is well-suited for solving combinatorial optimization problems, such as routing. Unlike NN, which follows a greedy approach and may get stuck in local optima, SA’s probabilistic nature enables it to explore a broader solution space, increasing the likelihood of finding a globally optimal route.

G. Different Approach

If I had to do this project over again, I would do the following:

Adjusting for speed: If I had varying speeds, I would have to rework my nearest neighbor algorithm for each truck to account for travel time.

Adjusting for weight: Because I only used 3 trucks and 40 packages, the weight was not evenly distributed. Had I manually loaded each package to achieve balance, I would most likely have reduced travel time.

Linear Probing (Chaining Hash Table): The hash table uses linear probing, which causes a longer search time. Quadratic probing increases search time and reduces data clustering.

H. Verification of Data Structure

The chaining hash table was able to meet all scenario requirements. The search method supports constant-time average retrieval of package data, enabling real-time tracking of package statuses as the scenario requires. The dynamic nature of the hash table allows for efficient handling of multiple packages (e.g., the 40 packages in the scenario) without performance degradation. The implementation supports updates to package details (e.g., correcting the address of Package #9 at 10:20 a.m.) through the insert method, which overwrites existing entries if the key already exists. Its efficiency and support for real-time operations align with the constraints of the delivery system, such as staying under the 140-mile limit and meeting deadlines.

H1. Other Data Structures

Two data structures are Priority Queues and Adjacency Lists

H1a. Data Structure Differences

Priority Queues:

A priority queue organizes elements based on priority levels, allowing access to the highest-priority element first. In the routing context, locations could be prioritized based on their proximity or delivery deadline. Unlike the NN algorithm, which uses a simple list or array to determine the following location, a priority queue would provide a more structured and dynamic approach to selecting the next optimal stop.

Adjacency Lists:

An adjacency list is a graph representation where each node stores a list of its neighboring nodes. It is memory-efficient and ideal for sparse graphs. Srishti Kumari (Kumari) states, “Adjacency lists can also be used to store and retrieve data. In addition, they can be used to implement algorithms on graphs. Like the shortest path or topological ordering, which requires numeric values.” For routing, an adjacency list could represent delivery locations and their distances. While a hash table focuses on key-value storage and retrieval, an adjacency list is designed explicitly for graph-based operations, making it more effective for solving routing problems involving interconnected locations.

I. Sources

baeldung. “Simulated Annealing Explained | Baeldung on Computer Science.” *Www.baeldung.com*, 26 Mar. 2023, www.baeldung.com/cs/simulated-annealing. Accessed 24 Jan. 2025.

Kumari, Srishti. "Introduction to Adjacency List for Graph." *Codedamn News*, codedamn, 13 Jan. 2023, codedamn.com/news/programming/introduction-to-adjacency-list-for-graph#advantages_of_an_adjacency_list. Accessed 24 Jan. 2025.

J. Professional Communication