

'''

## C950 WGUPS Project - Implementation Steps - Start

Please note; this is an example of implementation utilizing Nearest Neighbor Greedy Algorithm.

Feel free to use it by citing this document, however it is highly recommended that you come up with your own approach

---

### A) Package data steps:

1-Create HashTable data structure (See C950 - Webinar-1 - Let's Go Hashing webinar)

2-Create Package and Truck objects and have packageCSV and distanceCSV and addressCSV files ready

3-Create loadPackageData(HashTable) to

- read packages from packageCSV file (see C950 - Webinar-2 - Getting Greedy, who moved my data webinar)
- update Package object
- insert Package object into HashTable with the key=PackageID and Item=Package

### B) Distance data steps:

B.1) Upload Distances:

4-Create distanceData List

5-Define loadDistanceData(distanceData) to read distanceCSV file

- read distances from distanceCSV file; row by row
- append row to distanceData (two-dimensional list. See [C950 WGUPS Distance Table Matrix](#))

B.2) Upload Addresses:

6-Create addressData List

7-Define loadAddressData(addressData) to read addressCSV file

- read only addresses from addressCSV file
- append address to addressData.

### C) Algorithm to Load Packages:

C.1) Function to return the distance between two addresses:

8-Define distanceBetween(address1, address2)

9-Return distanceData[addressData.index(address1)][addressData.index(address2)]

i.e. distances between addresses can be accessed via `distanceData[i][j]`;

C.2) Function to find min distance/address:

10-Define `minDistanceFrom(fromAddress, truckPackages)`

11-Return min distance address to `fromAddress`

i.e. call `distanceBetween(address1, address2)` in a loop for all the addresses in the Truck

C.3) Function to load packages into Trucks:

12-Define `truckLoadPackages()`

13-Load Trucks based on assumptions provided (ex. Truck-2 must have some packages, some packages go together, some packages are delayed, ...)

14-And closest addresses/packages until there is 16 packages in a Truck

i.e. Load manually/heuristically or Loop package addresses and call `minDistanceFrom(fromAddress, truckPackages)` for all the addresses in the Truck not visited yet

#### **D) Algorithm to Deliver Packages:**

D.1) Function to deliver packages in a Truck:

15-Define `truckDeliverPackages(truck)`

16-Loop truck package addresses and call `minDistanceFrom(fromAddress, truckPackages)` for all the addresses not visited yet

D.2) Keep track of miles and time delivered:

17-Update delivery status and time delivered in Hash Table for the package delivered and keep up with total mileage and delivery times.

i.e. How to keep track of the time?:

`timeToDeliver(h) = distance(miles)/18(mph)` where 18 mph average Truck speed.

`time_obj = datetime.timedelta(hours=int(h), minutes=int(m), seconds=int(s))`. `time_obj` could be cumulated to keep track of time.

#### **E) UI to Interact with the Users:**

18-Create an UI to interact and report the results based on the requirements.

Possible Menu Options:

\*\*\*\*\*

1. Print All Package Status and Total Mileage

2. Get a Single Package Status with a Time
3. Get All Package Status with a Time
4. Exit the Program

\*\*\*\*\*

Possible output example:

PackageID, Address, City, State, Zip, Delivery Deadline, Mass KILO, PageSpecial Notes, Status, DeliveryTime

1, 195 W Oakland Ave, Salt Lake City, UT, 84115, 10:30 AM, 21, , ... Delivered by Truck-2, 08:46:20

2, 2530 S 500 E, Salt Lake City, UT, 84106, EOD, 44, , ... AtHub

3, 233 Canyon Rd, Salt Lake City, UT, 84103, EOD, 2, Can only be on truck 2, ... InRoute by Truck-2

...

...

...

C950 WGUPS Project - Implementation Steps - End

-----

""