Colab Link for Task-1: **Task-1(Hashing)**.ipynb

You have to create a hashTable from a vehicle_info array. The array contains vehicles as tuples and where each tuples indicates → (brand name, vehicle_type, rent, no_of_passengers)

Vehicle_info = [('Toyota', 'Private Car', 500, 4), ('Jeep', 'SUV', 950, 6), ('Lamborghini', 'SUV', 6900, 6), ('Hyundai', 'Bike', 100, 1), ('BMW', 'Private Car', 1000, 8), ('Honda', 'Bike', 150, 1), ('Ferrari', 'Private Car', 2500, 4), ('BMW', 'Minivan', 5800, 7)]

The print_vehicle_hashtable() methods have been done for you. Write the __hash_function() and insert_vehicle() methods.

def __hash_function(self, brand): 4 marks

The hash function takes the brand name as string calculates its hash key and returns the key. The hash key can be calculated by summing the ascii values of each character of the brand name and modding it by the length of the vehicle_info array. i.e. sum of ascii values%length of vehicle_info array.

[HAVE TO USE RECURSION]
[You may use helper function]

For example,
Brand = 'Toyota'
Ascii Values → T: 84, o:111, y:121, t:116, a:97
Sum of ascii values= 84+111+121+111+116+97= 640
sum of ascii values%length of vehicle_info array = 640%8 = 0 (hash key)

insert_vehicle(self, vehicle): 6 marks

You need to insert each vehicle from the **Vehicle_info** array into the hashtable on the basis of its hash function. If you want to insert a vehicle to an index of the hashTable where another vehicle info already exists, you have to check the brands of the vehicles. If the brand matches, then chain the current vehicle as a linked list in the hashed index. If it does not, find the earliest empty slot using the find_empty_slot() function and place the vehicle in that empty spot.
For example:

__hash_function('Jeep') will return 4.
0 : (Brand: Toyota, Type: Private Car, Rent: 500, No. of Passengers: 4)---->None
1: None
2: None
3: None
4: (Brand: Jeep, Type: SUV, Rent: 950, No. of Passengers: 6)---->None
5: None
6: None
7: None
8: None

__hash_function('Lamborghini') will return 4. Since index 4 has Jeep and the brands do not match, it goes to the earliest empty index 1.
0 : (Brand: Toyota, Type: Private Car, Rent: 500, No. of Passengers: 4)---->None
1: (Brand: Lamborghini, Type: SUV, Rent: 6900, No. of Passengers: 6)
2: None
3: None
4: (Brand: Jeep, Type: SUV, Rent: 950, No. of Passengers: 6)---->None
5: None
6: None
7: None
8: None

Again, for the 'BMW' vehicles, the chain is created since the brand is the same.

Colab Link for Task-2: **Task-2 (Tree)**.ipynb

Submit two colab files through the following submission Link within tomorrow(Tuesday) before the class(11 am):
Submission Link: **Submission Link**