**Lab # 3**

Due: 2024-10-11

By: Omar Abdul, Anas Taimah, Yusuf Khan

300228700, 300228842, 300293842

Course: CSI3120

Group: 40

**Code explanation:**

**Defining structures:** Our group started by defining the location structure which holds the location name, (X, Y) coordinates as well as the priority for each location. We also defined the vehicle structure which holds the vehicle id, capacity and its route.

**Helper function (distance):** We defined a helper function named distance which takes two locations and takes the difference between the x and y values of each location. With the new x and y value, the code applies Pythagorean theorem to find the distance.

**Helper function (split\_at):** The helper function we defined splits a list at a given index n and returns the first list with n elements and a second list with the rest of the elements. The base case when the given index is less than or equal to 0 will return an empty first list and the same input list as the second. If the input list is empty, it will return two empty lists. For non-empty lists, the function splits the head off the list then recursively calls the function with n-1 on the remaining part of the list. The function will then return the new split lists.

**Step 1 Input Delivery Locations:** Depending on how many locations are being asked for (n), if it is 0 an empty list is returned. Otherwise, for every n, it asks the user to fill in its location details such as the name, coordinates and priorities. The function will then return a list of these records once every detail has been inputted.

**Step 2 Input Vehicle Details:** After finding out the number of vehicles (n), the function will ask the user details of every vehicle. If n = 0, no details will be asked, otherwise the function recurses through every vehicle assigning it a vehicle id and an empty route as well as asking the user for its capacity. A list of all vehicle records are then returned at the end.

**Step 3 P1 Sort Locations by Priority:** This function takes the delivery location list and sorts them based on priority in descending order (3,2,1).

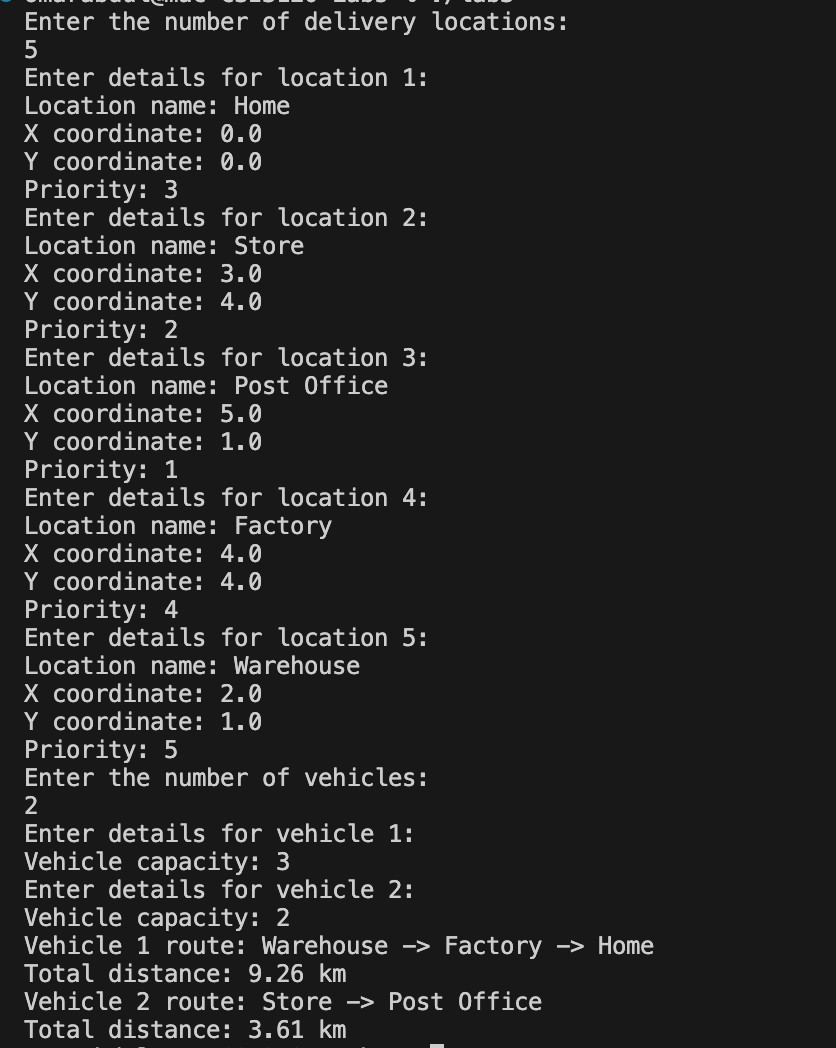
**Step 3 P2 Assignment locations to vehicles:** This function defines a separate recursive function called assign which takes the locations and vehicles and returns a list with locations assigned to vehicles. It uses the split\_at helper function to add as many locations as possible to each vehicle. The recursive function will end once no elements remain in either of the location or vehicle list.

**Step 3 P3 Calculate route distances:** This function takes all the locations a vehicle will travel and return the total distance the vehicle will travel. The base case is when only 0 or 1 elements remain in the input list. The function recurses through the list and uses the distance helper function on the first and second element in the list. After the distance is found and added to the accumulator, it recurses without the first element.

**Step 4:** This function displays each vehicles route as well as its distance.

**Main:** The main function first asks the user for the number of delivery locations, then calls input\_locations to get each locations details. It then asks the user for the number of vehicles, then calls input\_vehicles to get each vehicles details. After calling the sort\_locations\_by\_priority to sort the locations by priority it calls assign\_locations\_to\_vehicles. Finally, the main function calls the display function to display all the results.

**Test cases:**



**ChatGPT declaration:** Our group used ChatGPT throughout the lab to better understand the program application so that we are able to develop our answers. ChatGPT was used for assistance to learn and understand how it worked and not to generate any answers or results. All solutions were implemented in our own words and approach.