Problem 2 approach

Goal: Maximising the number of packages in each delivery, estimating the time for each package to be delivered.

- Define the packages and vehicle objects
    - For packages, weight, distance are key factors (cost/discount previously calculated and not necessary for what problem is actually asking)
    - For vehicles availability is a key factor, take into account delivery time when considering availability. Max speed & weight limit are given constants
- Find the selected packages for the delivery given that they meet all the criteria (combinations)
    - Maximise number of packages in each delivery (primary concern) so that the total is under the weight limit
    - If number of packages are the same, maximise the weight of the delivery
    - If the number of packages are the same and the weight is the same, the shortest delivery distance should be used
- While packages are available, deliver the packages
    - A vehicle must be available for the packages to be delivered
- Record the time for each package to be delivered
- Estimate the return trip by taking the maximum distance of the package combination
- Calculate the total time for delivery (considering the return trip)
- Continue for each package combination, given that vehicles are available
- If a vehicle is not available, estimate time until the vehicle returns (from total time for delivery) and keep a record of the time.

- Work out which vehicle arrives first, as that will be the next vehicle out for delivery
    - Use that vehicles time until available to add to the recorded time for delivery for next deliveries to receive total time for delivery
- Repeat until there are no packages left

For each part:

- Defining: python classes (from problem 1)
- Selected packages: python combinations function with conditional operators to return the best combination that meets all criteria
- Deliver the packages; loop through the process until packages are not available.
    - Initialise an empty array package_deliveries to store the records of the delivered packages
    - While packages are available (packages in packages_array, not the array above):
    - Use the next available vehicle to deliver packages by choosing the soonest available vehicle
    - Calculate the delivery time per package
    - Calculate the delivery cost (for given sample outputs, not vital to question, taken from problem 1)
    - Append pkg id and time for delivery (and delivery cost/discount applied) to the empty array, time for delivery should account for waiting time of the vehicle (first two runs should be delivery time + 0)
    - Take the maximum distance of the delivery and calculate delivery return time
    - Add the delivery return time to vehicle available after for the next iteration
    - remove the delivered packages from the available packages array

- Within the process consider waiting time for the vehicles
- If both vehicles are out then no packages can be delivered until accounting for previous delivery time

- Run delivery and iterate through the records to print the output