

Delivery company management

A delivery company (located in Algiers) is asking for a technical solution to automate and manage its services.

An item to be delivered is characterized by its **identifier** (unique integer), its **entry date**, the **wilaya** of delivery (integer from 1 to 58), its **weight** and its **status** (awaiting delivery, returned).

The company has two types of vehicles in its fleet: **motorcycles** and **vans**. Each vehicle is characterized by an **identifier** (unique integer) and a **maximum capacity** (number of items it can carry).

The company uses two text files **Items.txt** and **vehicles.txt** (ordered according to identifiers) in which information about items and vehicles is stored: one element per line, with fields separated by spaces.

To manage deliveries, we propose to use linked lists and queues as follows: **ItemL** list that stores the items to be delivered, **VehicleL** list for vehicles, and two queues: one for motorcycles, **MotoQ**, and another for vans, **VanQ**. The **ItemL** and **VehicleL** lists are created from the **Items.txt** and **vehicles.txt** files, and the **MotoQ** and **VanQ** queues are created from the **VehicleL** list.

You are asked to:

- Create the different lists and queues models and get information loaded into lists and queues.
- Add a new item to the **ItemL** list (new delivery item)
- Add a new vehicle to the **VehicleL** list or delete a vehicle. These actions imply also the modification of the corresponding queue.
- Simulate the delivery operation:
 - Items assigned to a vehicle are represented by a list linked to the corresponding node in its queue.
 - Priority is given to items with the earliest entry date.
 - Motorcycles do not make deliveries outside of Algiers, do not take items weighing more than 3kg and can only carry 2 items at once.
 - Vans deliver outside Algiers but can only deliver to one wilaya at once.
 - Deliveries are launched as vehicles reach their capacity or after all items in **ItemL** have been processed.
 - A vehicle leaving for delivery is dequeued from the corresponding queue.

Note: A vehicle can leave (is dequeued) only if it is at the head of the queue. A vehicle in the middle of the queue must therefore wait for the vehicles in front of it to leave.

- Simulate the comeback of a vehicle
 - Each item is either delivered or returned. A delivered item is removed from the **VehicleL** list.
 - The vehicle is added to the tail of the corresponding queue. After 3 trips on the same day, a vehicle is rendered out of service until the next day.
- Simulate the returns pick-up (by the sender) or delivery cancellation, the item is removed from the **VehicleL** list.
- Simulate the end of the day
 - Update items and vehicles files (build new files from lists **ItemL** and **VehicleL**, no processing will be done on the original files)
 - Generate end-of-day reports: number of items delivered, of returns, of new items...

Note

- This work must be done by two students
- The code must be written in C, and must be clear and commented.
- The data structure models and implemented functions must be included in a library.
- Work that does not use the models (abstract machines) will not be examined.
- Submit a compressed project named: Name1_Name2_Gxx.
- **Due date: 07/04/2024**