# PROGRAMMING TASK

This assignment involves creating a Java program that simulates cargo delivery between distribution centers for a shipping company. You will be given a set of cities, each with a distribution center with cargo packages and delivery vehicles. You must complete several missions, each involving building a simulated delivery vehicle and loading it with cargo packages at a certain city. Then, you must follow a specific route to reach the destination city, dropping off and picking up cargo packages along the way. Your Java program should read the input files that contain information about the cargo packages and generate an output file that shows the final status of the distribution centers in each city.

For this assignment, you **must** use **your own implementations** of *doubly linked list*, *stack* and *queue* by taking inspiration from your textbook. They should use generics and use a linked-list based implementation. You are **not allowed** to use any **external library** or **.jar file**.

Your program should simulate **cargo delivery** based on input files and generate an output file. The cities.txt file contains information about the cities that are possible destinations for your simulations. Each city has a distribution center with cargo packages and delivery vehicles. The cargo packages are delivered and transported according to the **stack principle** (LIFO). The packages.txt file shows the initial state of the cargo packages in each city. The delivery vehicles are stored in a central warehouse at the distribution center and follow the **queue principle** (FIFO). The vehicles.txt file shows the initial state of the delivery vehicles in each city. The **delivery vehicle and its cargo packages** are represented by a doubly linked list.

The vehicle class inherits from the cargo packages class, so you will have doubly linked lists of cargo packages as your carriage. The missions.txt file contains the missions that you have to complete.

This is how a mission is structured: X-Y-Z-a,b-c1,c2,...

* X: Source distribution center
* Y: Middle distribution center
* Z: Destination distribution center
* a: Number of cargo packages you load from the source distribution center
* b: Number of cargo packages you load from the middle distribution center
* c: Cargo package indices that you must drop off at the middle distribution center. Indices start at 0.

In order to finish a mission, you must follow these steps:

1. **Assemble a vehicle at the starting city (X)**: The first delivery vehicle available in the central warehouse at the source city must be placed at the head of your list because a delivery vehicle is modeled as a **doubly linked list** of cargo packages. After that, you need to add the cargo package(s) from city **X** to this list.

1. **Add the b cargo packages from the middle city (Y)** to your list.

1. **Drop off cargo packages c1, c2, ... at the middle distribution center (Y)**.

1. **Place the vehicle and remaining cargo packages at the destination city (Z)**, disbanding the cargo packages upon arrival.

You must be aware of the sequence in which insertions and removals are made during a mission.

The four input files—cities.txt, packages.txt, vehicles.txt, and missions.txt— will be used as parameters by your Java program. Reading the input from these files will yield the distribution centers' initial configuration. Following reading the mission, the delivery vehicle is chosen, and the mission's cargo packages are distributed. After the mission is finished, the status of every distribution center should be recorded/written in the result.txt output file.

Here are the four sample input files (under set2 inside sampleIO folder on LMS) content:

|  |
| --- |
| **cities.txt packages.txt** Berlin P1 Berlin  Hamburg P2 Berlin  Munich P3 Hamburg  P4 Berlin  P5 Hamburg  P6 Munich  P7 Munich  P8 Hamburg  P9 Berlin  P10 Hamburg    **vehicles.txt missions.txt**  V1 Hamburg 4.8 Berlin-Hamburg-Munich-3-2-1,2  V2 Berlin 7.2  V3 Berlin 2.8  V4 Hamburg 4.3 |

The output file, result.txt content for the above sample files run is given below. Please check your program with these input files as well as the others that are already shared together with the homework files on LMS.

Berlin Packages: P1

Vehicles:

V3

-------------

Hamburg Packages:

P2

P4

P5

P3

Vehicles:

V1

V4

-------------

Munich Packages:

P8

P10

P9

P7

P6

Vehicles: V2

-------------

Sample console usage of your program:

% java Main cities.txt packages.txt vehicles.txt missions.txt result.txt