

The Rise of Geometrica's Courier System - MILESTONE 02



After the remarkable success of the **Triangle Optimization Project**, the town of **Geometrica** saw a significant improvement in delivery times. The newly optimized triangular routes allowed couriers to navigate the town with greater efficiency, cutting down travel time and boosting productivity. The citizens rejoiced, businesses flourished, and Geometrica's reputation as a leader in logistics was restored.

However, just as the town was beginning to enjoy its newfound efficiency, a **new problem emerged**. While couriers could now travel along optimized routes, the **post offices themselves** were struggling to keep up with the rising demand for deliveries. Some post offices were overflowing with packages, while others sat nearly empty, leading to widespread delivery delays and an imbalance in workload distribution.

At an emergency town hall meeting, **Mayor Vertex** addressed the concerned citizens: "We have solved the problem of inefficient routes, but our work is far from over. The **Post Transition System**, responsible for managing packages across post offices, is now facing a critical challenge. If we do not redistribute our packages efficiently, all our hard work will be undone. We must act quickly to balance the load across all post offices before our courier system collapses."

The Town Council has called upon the **Triangle Optimization Task Force** once again. Your new mission is to develop an **automated system** that will redistribute packages efficiently across Geometrica's post offices while ensuring that each post office operates **within its weight constraints**. This system, known as the **Package Redistribution Plan**, is the final piece in restoring balance to the courier system.

Objective

Your task is to design and implement a **Post Transition System** that automates the process of transferring packages between post offices. The system must adhere to the following key principles:

1. **Package Listing:** Display all packages in a town's post offices.
2. **Package Transfer:** Move packages from one post office to another while ensuring that only packages within the target post office's **weight constraints** are accepted.
3. **Load Monitoring:** Identify the town with the **most packages** after transfers are completed.

This system will ensure that **no post office is overwhelmed with packages** while others remain underutilized. The council is counting on your team to develop a fair and **efficient redistribution strategy**.

Challenges in the Post Transition System

1. **Weight Constraints:** Each post office has a **minimum weight** and **maximum weight** that it can handle. Packages outside this range **cannot** be accepted. Your system must respect these constraints.
2. **Limited Transfer Options:** Not all packages will have an eligible post office to move to. Your system must ensure that packages are only moved **if a suitable destination exists**.
3. **Dynamic Updates:** Each time a transfer is made, the package lists in both the **source** and **destination** post offices must be updated in real-time.

To solve these problems, your system will process package transfers efficiently while keeping track of town-wide package distributions.

Input Format

The input consists of:

1. An integer t , representing the number of divisions.
 2. For each division:
 - o A string name, the name of the division.
 - o An integer O , the number of post offices in the division.
 - o For each post office:
 - Two integers min_weight and max_weight, representing the **weight range** of packages that the post office can handle.
 - An integer p , the number of packages currently stored in the post office.
 - For each package:
 - A string id, the unique package identifier.
 - An integer weight, the weight of the package.
3. Then, there is a single integer q on the line, which is the number of operations. operation blocks follow, each describing an operation. Every operation block contains several lines. On the first line, there is an integer 1, 2, or 3. If this integer is 1, the next line contains a single string division_name, representing the name of the division for which all packages should be printed. If the integer is 2, the next line contains the strings source_name and target_name, followed by the integers source_office_index and target_office_index, separated by single spaces. This indicates a transfer of packages between post office source_office_index in the division source_name and post office target_office_index in the division target_name. If the integer is 3, no additional lines follow, and the division with the most packages should be determined.

Output Format

For each operation performed, the system should generate structured output as follows:

1. Operation 01: Listing Packages in a Division:

Division: [Division Name]

Post Office [Index]:

[Package ID 1] [Package ID 2] ...

Post Office [Index]:

[Package ID 1] [Package ID 2] ...

2. Operation 03: Finding the Division with the Most Packages:

Division with the most number of packages is [Division_Name]

The Council's Expectations

Mayor Vertex issues a final directive:

“Our work is not complete until every package is accounted for. By balancing the package loads across all post offices, we will ensure that deliveries happen smoothly, couriers remain efficient, and our citizens receive their goods on time. The success of this system will determine the future of Geometrica’s economy.”

Your team must develop a **flawless system** that redistributes packages in real time, ensuring **no post office is overloaded** while maintaining operational fairness.

Why This Matters

1. Efficient Delivery Operations:

A well-balanced post office system ensures that **no single location is overburdened** while others remain underutilized. This allows couriers to **deliver packages faster**.

2. Economic Growth:

Local businesses depend on **quick and reliable deliveries**. An optimized post office system ensures that businesses can function **without disruption**.

3. Sustainability & Reduced Wastage:

A balanced system will **reduce unnecessary trips**, lowering fuel consumption and making Geometrica’s **courier system eco-friendly**.

Are You Ready to Restore Balance? 😎

Geometrica has come a long way in its journey to **logistics excellence**, but this final step will determine whether the town can maintain its **newfound efficiency**. As a member of the **Triangle Optimization Task Force**, you are **Geometrica’s last hope**.

The fate of the town’s **courier network** rests in your hands. Can you design the perfect **Post Transition System** to restore balance and ensure **smooth operations** for generations to come?

Sample Test Cases

Test Case 01:

Input	Output
2 M 1 3 2 6 m1 2 m2 3 m3 6 N 2 2 1 4 n1 1 n2 3 3 2 5 n3 2 n4 4 n5 5 5 3 2 N 1 M 0 3 1 M 1 N	Division with the most number of packages is N Division with the most number of packages is M M: 0: m1 m2 m3 n3 n4 n5 N: 0: n1 n2 1:

Test Case 02:

Input	Output
2 X 2 2 1 5 p1 3 p2 5 1 2 4 p3 2 Y 1 3 2 6 p4 2 p5 4 p6 6 4 3 2 Y 0 X 1 3 1 X 1 Y	Division with the most number of packages is X Division with the most number of packages is X X: 0: p1 p2 1: p3 p4 p5

Submission

Marks for milestone 2

10% of the final grade (zero marks for submissions that have compilation errors)

Deadline for milestone 2

Strictly on or before March 9th 2025 at 11.55 pm.

Plagiarism policy

Copied submissions (including those from the Internet) will receive zero marks. Your program must be entirely your own work. **Do not copy from others, and do not allow anyone else to see your code.**

Submission files

For Milestone 2, you need to submit your code through VPL under “Project - Milestone 02,” named Project Milestone 02. Note that marks will also be awarded for writing legible and readable code with proper comments. **Late submissions won’t be accepted.**

In addition to the code, you must submit **a report of no more than two pages, with one-inch margins on all sides**. In this report, you should explain how you approached and solved the problem, rather than simply describing the code itself. **The report must be written in your own words, and AI detection must show 0% AI-generated content.** If any AI-generated content is detected, you will receive zero marks for the entire milestone. **Late submissions will be accepted with a 20% penalty per day they are late.**

