Problem Description:

Suppose a company has several factories that produce a certain product, and several warehouses that distribute the product. Each factory can produce a certain amount of the product, and each warehouse has a certain demand for the product. The goal is to minimize the cost of transporting the product from the factories to the warehouses, subject to the constraints of supply and demand.

Write a Python program to implement the Northwest Corner method for solving the transportation problem. Your program should take the following inputs:

- A list of factories, each represented by a tuple of the form **(supply, [costs])**, where **supply** is the amount that the factory can produce, and **costs** is a list of transportation costs from the factory to each warehouse. The length of **costs** should be equal to the number of warehouses.
- A list of warehouses, each represented by a tuple of the form **(demand, [costs])**, where **demand** is the amount that the warehouse needs, and **costs** is a list of transportation costs from each factory to the warehouse. The length of **costs** should be equal to the number of factories.

Your program should return a list of shipments, where each shipment is represented by a tuple of the form **(factory, warehouse, amount)**, indicating that **amount** units of the product should be shipped from **factory** to **warehouse**.

Your program should implement the Northwest Corner method as described above. If there are multiple optimal solutions, your program should return any one of them.

Sample Input:

```
factories = [(10, [2, 3, 1]), (20, [4, 5, 6])]

warehouses = [(8, [5, 4, 3]), (15, [2, 3, 1]), (5, [4, 1, 2])]
```

Sample Output:

```
[(0, 0, 8), (0, 1, 2), (1, 1, 13), (1, 2, 5)]
```