

### ปัญหา ขนส่งสินค้า 4 (Product Shipping 4)

จากข้อมูลคลาสรถขนของ (Truck) ในข้อ ProductShipping3 ถ้ารถขนของมีการอัปเกรด สินค้าจะถูกส่งไวขึ้น ซึ่งจะคิดค่าธรรมเนียมการส่งเป็น 2 เท่าของราคาปกติ

\*หมายเหตุ ค่าธรรมเนียมปกติ คิดจาก (น้ำหนักเป็นกิโลกรัม x ระยะทาง)

ตัวอย่างเช่น น้ำหนักสินค้า 28500 กรัม ส่งไปปลายทาง 8

คิดค่าขนส่งแบบพิเศษเป็น  $2 \times (28500/1000) \times 8 = 456$  บาท

จงสร้างคลาสรถขนของพิเศษ (SpecialTruck) สืบทอดมาจากคลาสรถขนของ Truck สร้างเมธอด setFee โดยเขียนทับ (Overriding Method) คำนวณค่าธรรมเนียมการส่งเป็น  $2 \times$  น้ำหนักเป็นกิโลกรัม x ระยะทาง

รูปแบบข้อมูลนำเข้า

บรรทัดที่ 1 จำนวนเต็ม 1 จำนวน คือจำนวนรถขนของ ( $0 < M < 100$ )

บรรทัดที่ 2 มีจำนวนเต็มทั้งหมด M ตัว แต่ละตัวคั่นด้วยช่องว่างเป็นน้ำหนักที่รถแต่ละคันสามารถขนได้

บรรทัดที่ 3 มีจำนวนเต็ม [1,2] ทั้งหมด M ตัว โดยเลข 1 แทนรถขนของปกติ และ 2 แทนรถขนของพิเศษ

บรรทัดที่ 4 จำนวนเต็ม 2 จำนวน คือจำนวนสินค้า ( $0 < N \leq 100$ ) และปลายทางเป็นตัวเลข ( $0 < D \leq 100$ )

อีก N บรรทัด แต่ละบรรทัด บอกข้อมูล 4 อย่างคือ ชื่อสินค้า น้ำหนักสินค้า ราคาสินค้า จำนวนสินค้า

รูปแบบผลลัพธ์

บรรทัดที่ 1 โดยแสดงหมายเลขรถขนของที่ทำการขนส่งสินค้าในใบสั่งซื้อนี้ได้ หรือแสดง “No truck” ถ้าไม่มีรถ

บรรทัดที่ 2 แสดงค่าขนส่งแบบปกติ และค่าขนส่งแบบพิเศษ

ตัวอย่างข้อมูลเข้าและผลลัพธ์

ข้อมูลเข้า	ผลลัพธ์
3 3500 2 1800 1 4000 2 2 3 orange 500 60 5 toy 100 80 10	Truck#1 Fee=21.0
3 1500 2 800 2 2000 1 2 3 orange 500 60 5 toy 100 80 10	No truck Fee=10.5 21.0

ส่วนโค้ดที่ใช้ในโจทย์

```

import java.util.Scanner;
class Product {
    //from ProductShipping1
}
class PurchaseOrder{
    //from ProductShipping2
}
class Truck {
    // from ProductShipping3
}
class SpecialTruck {
    // write your code here
}

```

```
public class ProductShipping4 {
    public static void main(String[] args){
        Scanner in = new Scanner(System.in);
        int m = in.nextInt();
        Truck[] trucks = new Truck[m];
        for (int i=0;i<m;i++){
            int capacity = in.nextInt();
            int type = in.nextInt();
            if (type==1)
                trucks[i] = new Truck(capacity);
            else
                trucks[i] = new SpecialTruck(capacity);
        }
        int n = in.nextInt();
        int dest = in.nextInt();
        Product[] products = new Product[n];
        int[] amount = new int[n];
        for(int i=0;i<n;i++){
            String name = in.next();
            int weight = in.nextInt();
            int price = in.nextInt();
            products[i] = new Product(name,weight,price);
            amount[i] = in.nextInt();
        }
        PurchaseOrder po = new PurchaseOrder(n,products, amount);
        int minvalue = Integer.MAX_VALUE;
        int minIndex = -1;
        for (int i=0;i<m;i++){

if((trucks[i].netWeight>=po.weight)&&(minvalue>trucks[i].netW
eight)){
            minvalue = trucks[i].netWeight;
            minIndex = i;
        }
    }
    if (minIndex >= 0){
        trucks[minIndex].add(po, dest);
        System.out.println("Truck#" + (minIndex+1));
        System.out.println("Fee=" + trucks[minIndex].getFee());
    }else{
        System.out.println("No truck");
        double fee = po.weight*dest/1000.0;
        System.out.println("Fee=" + fee + " " + (2*fee));
    }
}
}
```