



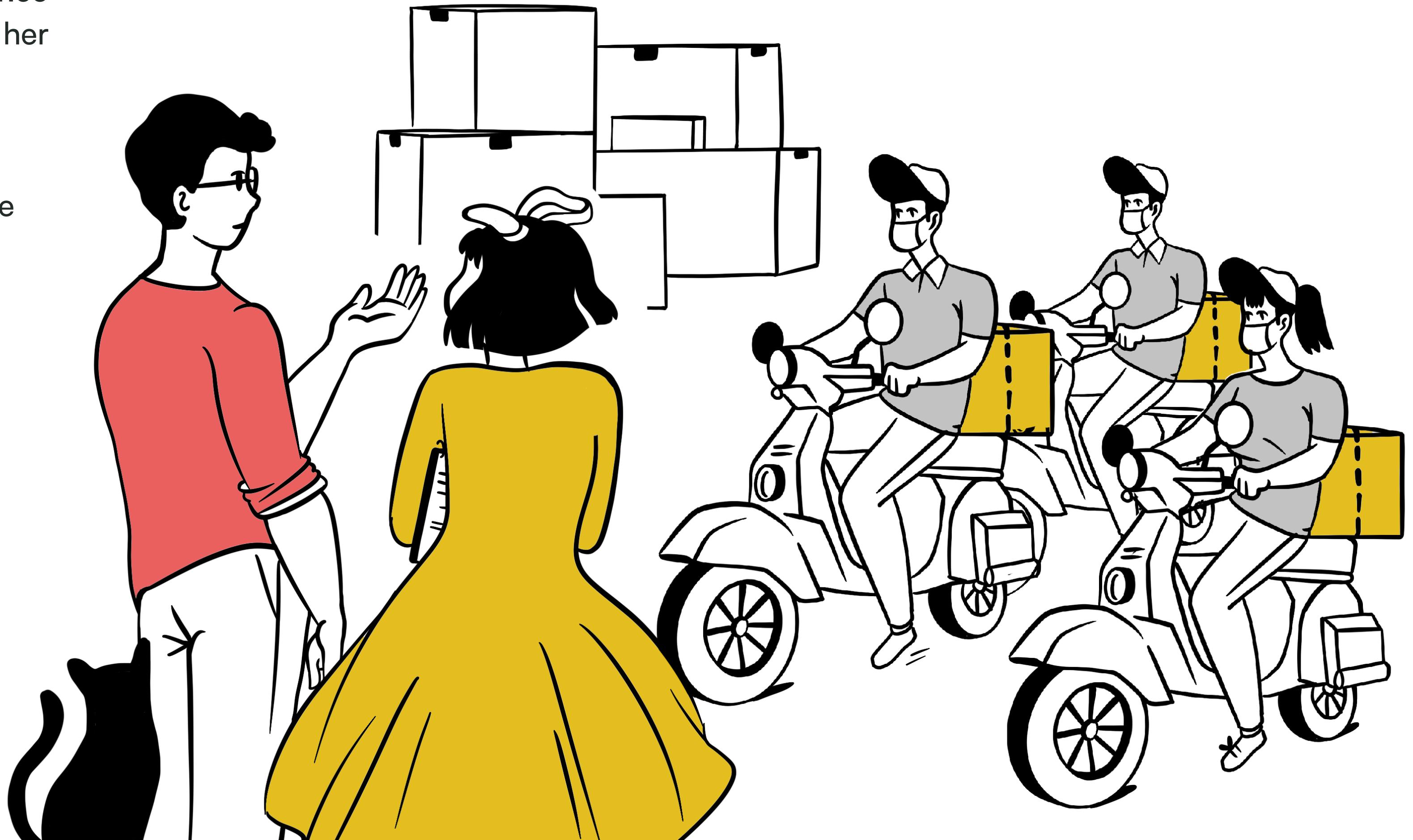
Craft what **you love**,  
**love what you craft**

**EVEREST** ENGINEERING

# Background

Kiki, a first-time entrepreneur from the city of Koriko has decided to open a **small distance courier service** to deliver packages, with her friend Tombo and cat Joji.

Kiki has invested in **N** no. of **vehicles** and have **driver partners** to drive each vehicle & deliver packages.



# Problem<sup>01</sup>

## Delivery Cost Estimation with Offers

Since it's a new business, the team has decided to pass coupons around the town which will help them attract more customers.

Things to keep in mind:

Only **one** offer code can be applied for any given package.

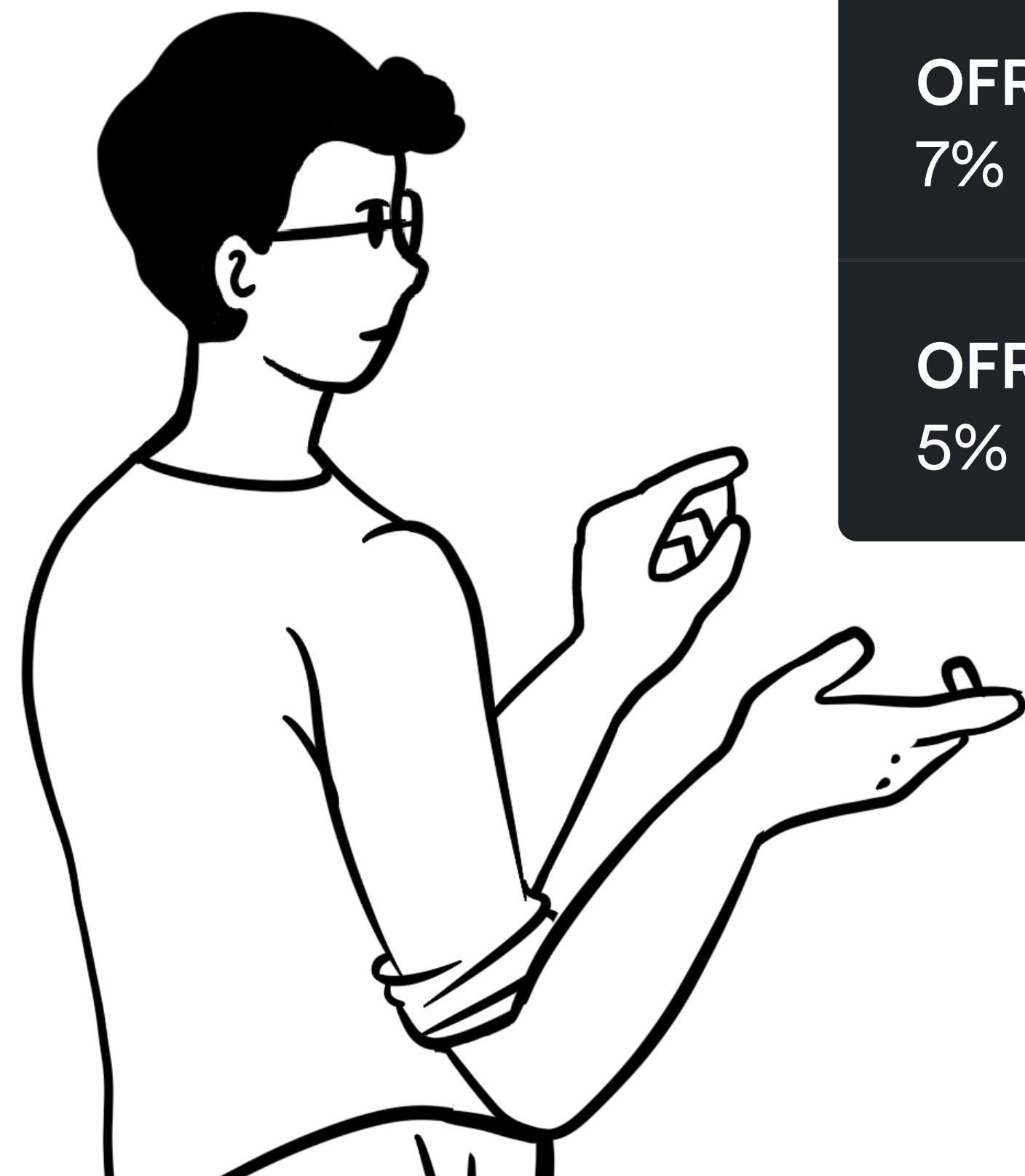
Package should meet the required mentioned **offer criterias**.

If offer code is **not valid/found**, discounted amount will be equal to 0.



# Offer Criteria

The offers can be used according to the criteria that Tombo has captured in this table. Discount will not be provided if no offer code is applied.



	Distance (km)	Weight (kg)
OFR001 10% Discount	< 200	70-200
OFR002 7% Discount	50-150	100-250
OFR003 5% Discount	50-250	10-150

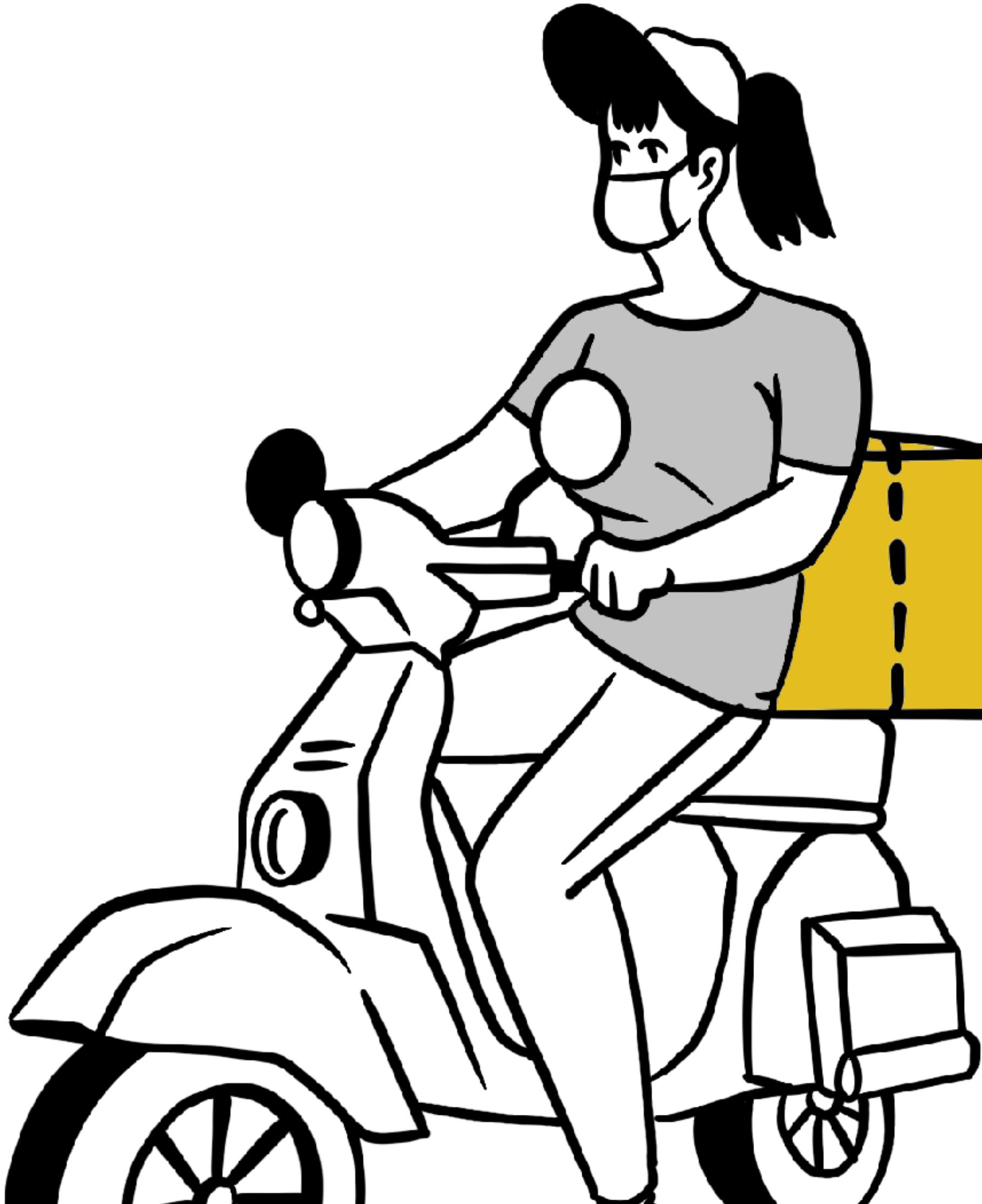
# Challenge

You are required to build a command line application to estimate the **total delivery cost** of each package with an offer code (if applicable).

Note: Code should be extensible & scalable for more offer codes.

i Base Delivery Cost + (Package Total Weight \* 10) +  
(Distance to Destination \* 5) =

Delivery Cost



# Sample Input Formats

## Input Format

base\_delivery\_cost no\_of\_packges  
pkg\_id1 pkg\_weight1\_in\_kg distance1\_in\_km offer\_code1  
...  
...

## Output Format

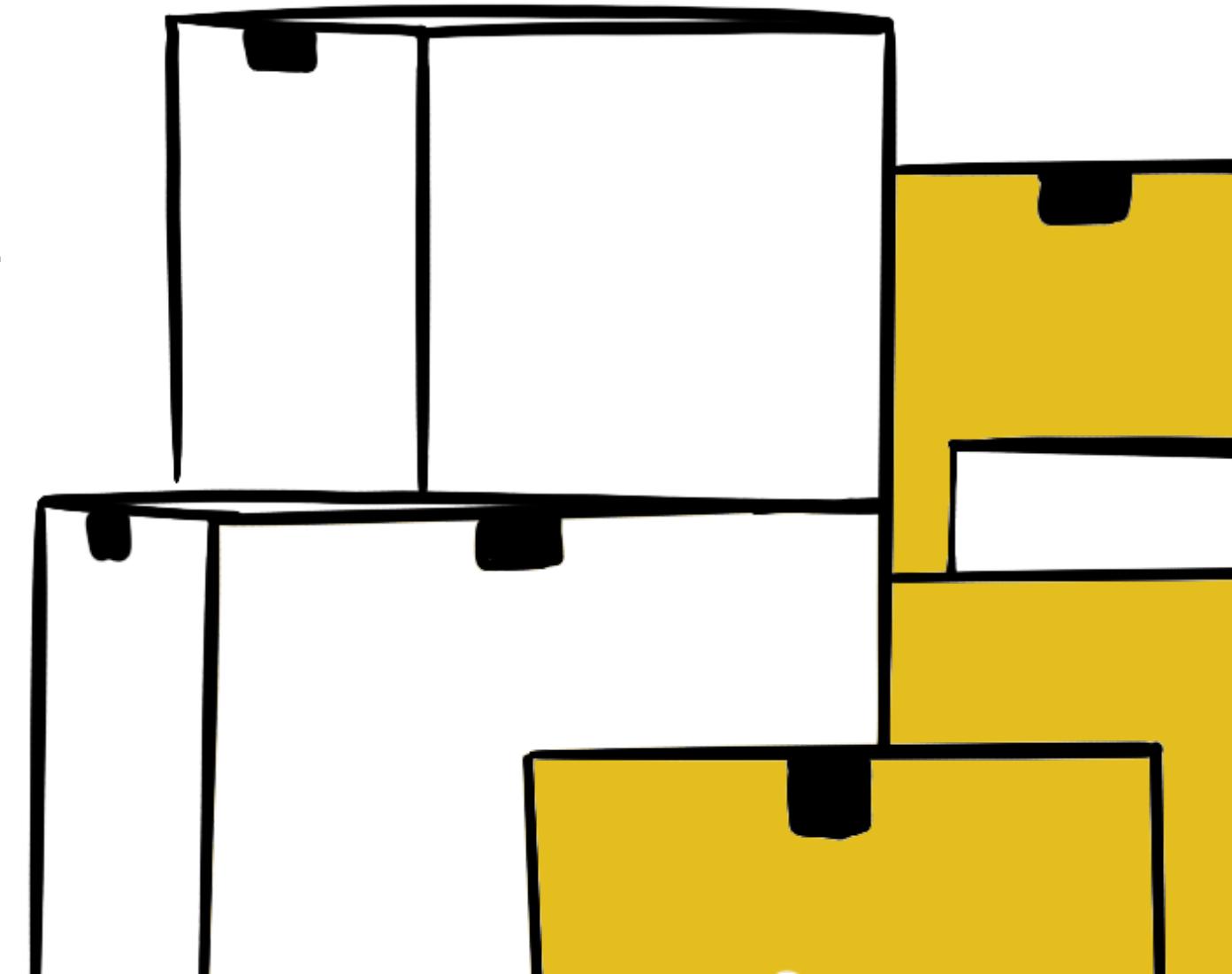
pkg\_id1 discount1 total\_cost1  
...  
...

## Sample (I)

100 3  
PKG1 5 5 OFR001  
PKG2 15 5 OFR002  
PKG3 10 100 OFR003

## Sample (O)

PKG1 0 175  
PKG2 0 275  
PKG 35 665



# Explanation

## PKG 1: Testcase

Base.delivery cost: 100

Weight: 5kg | Distance: 5km

Offer code: OFR001

<b>Delivery Cost</b>	<b>175.00</b>
$100 + (5 * 10) + (5 * 5)$	

<b>Discount</b>	<b>- 0.00</b>
(Offer not applicable as criteria not met)	

<b>Total cost</b>	<b>175.00</b>
-------------------	---------------

## PKG 3: Testcase

Base.delivery cost: 100

Weight: 10kg | Distance: 100km

Offer code: OFR003

<b>Delivery Cost</b>	<b>700.00</b>
$100 + (10 * 10) + (100 * 5)$	

<b>Discount</b>	<b>- 35.00</b>
(5% of 700 i.e; Delivery Cost)	

<b>Total cost</b>	<b>665.00</b>
-------------------	---------------

# Problem<sup>02</sup>

## Delivery Time Estimation

Now all these packages should be delivered to their destinations in the fleet of vehicles Kiki owns. She has N no. of vehicles available for delivering the packages.

Things to keep in mind:

Each Vehicle has a **limit (L)** on **maximum Weight (kg)** that it can carry.

All Vehicles travel at the same **speed (S km/hr)** and in the **same route**. It is assumed that all the destinations can be covered in a single route.



# Delivery Criteria

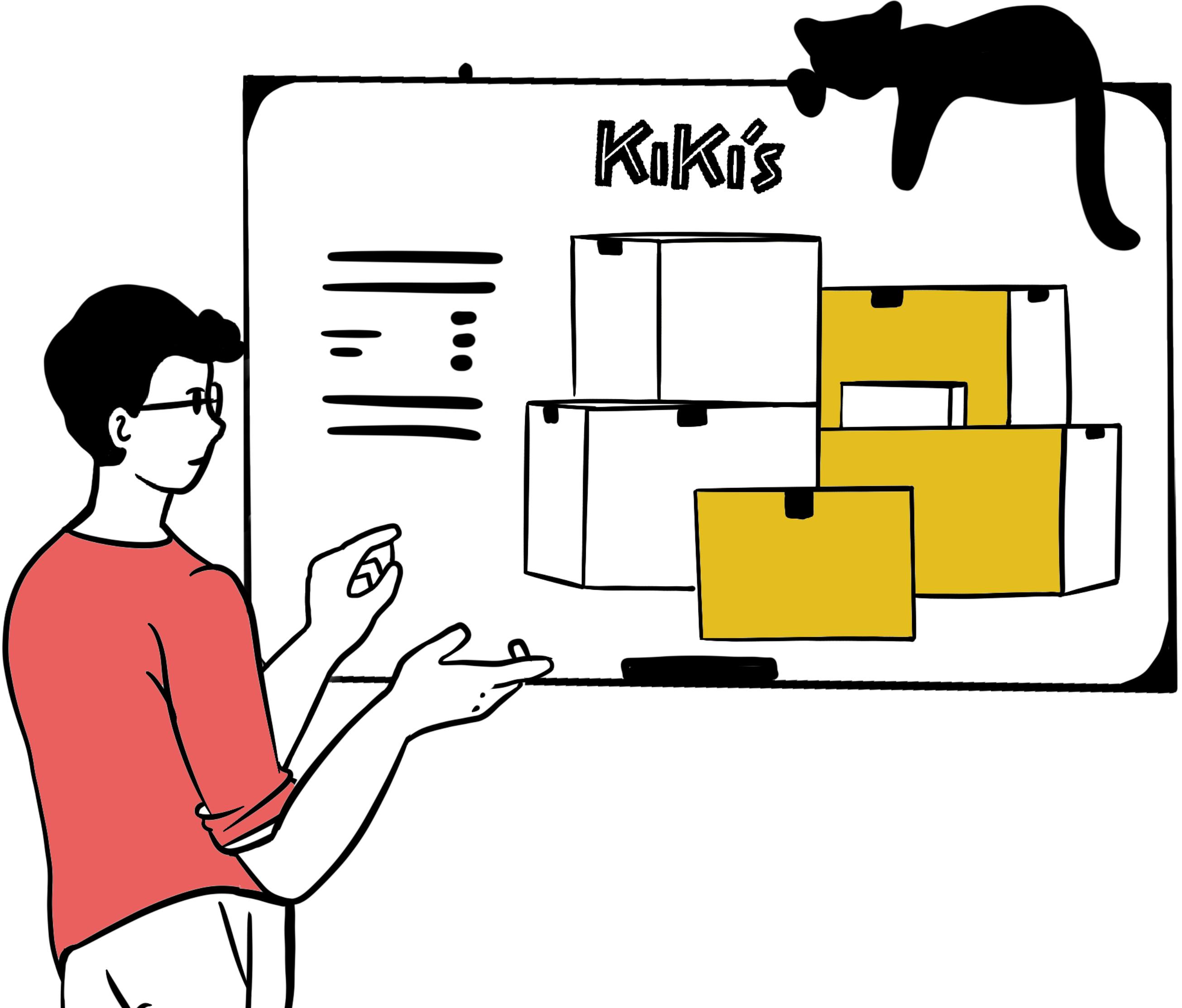
The delivery can be made using the criteria illustrated below. We should prefer heavier packages when there are multiple shipments with the same number of packages.



# Challenge

Tombo, the geeky business partner obsessed with delivery metrics wants to maximise efficiency.

You are required to build a command line application to calculate the **estimated delivery time** for every package by maximizing no. of packages in every shipment.



# Sample Input Formats

## Input Format

```
base_delivery_cost no_of_packges  
pkg_id1 pkg_weight1_in_kg distance1_in_km offer_code1  
....  
no_of_vehicles max_speed max_carryable_weight
```

## Sample (I)

```
100 3  
PKG1 50 30 OFR001  
PKG2 75 125 OFFR0008  
PKG3 175 100 OFFR003
```

```
PKG4 110 60 OFFR002  
PKG5 155 95 NA  
2 70 200
```

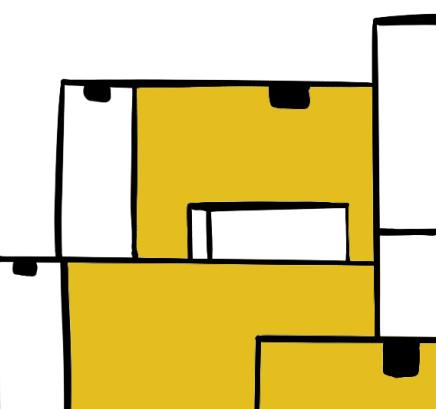
## Output Format

```
pkg_id1 discount1 total_cost1 estimated_delivery_time1_in_hours  
...
```

## Sample (O)

```
PKG1 0 750 0.42  
PKG2 0 1475 1.78  
PKG3 0 2350 1.42
```

```
PKG4 105 1395 4.417  
PKG5 0 2125 4.19
```



# Explanation

Packages	05	Vehicles	02	Max Speed	70km/hr	Max load	200kg
----------	----	----------	----	-----------	---------	----------	-------

## STEP 01

Packages Remaining: 05

Vehicles Available: 02 | Current Time: 0 hrs

PKG1 + PKG2 ⊕ 02 packages  
50kg      75kg      125kg

**Vehicle 01**      Delivering PKG1      **0.42 hrs**

Delivering PKG2      **1.78 hrs**

**Vehicle 01 will be available after (2\*1.78)**      **3.56 hrs**

## STEP 02

Packages Remaining: 03

Vehicles Available: 01 | Current Time: 0 hrs

PKG3      PKG4      PKG5 ⊕ 01 package (PKG3)  
175kg      110kg      155kg      175 kg (Most Weight)

**Vehicle 02**      Delivering PKG5      **1.42 hrs**

**Vehicle 02 will be available after (2\*1.42)**      **2.84 hrs**

# Explanation (Contd. 1)

## STEP 03

Packages Remaining: 02

Vehicles Available: 0 | Current Time: 0 hrs

Vehicle 01	Returning in	<b>3.56 hrs</b>
------------	--------------	-----------------

Vehicle 02	Returning in	<b>2.84 hrs</b>
------------	--------------	-----------------

Vehicle 02 will be available first after	<b>2.84 hrs</b>
--	-----------------

## STEP 04

Packages Remaining: 02

Vehicles Available: 01 | Current Time: 2.84 hrs

PKG4 110kg	PKG5 155kg	⊕	01 package (PKG5) 155 kg (Most Weight)
---------------	---------------	---	---

Vehicle 02	Delivering PKG5 (2.84+ 1.35)	<b>4.19 hrs</b>
------------	------------------------------	-----------------

Vehicle 02 will be available after (2.84+ 2*1.35)	<b>5.54 hrs</b>
---	-----------------

## Explanation (Contd. 2)

### STEP 05

Packages Remaining: 01

Vehicles Available: 0 | Current Time: 2.84 hrs

---

**Vehicle 01**      Returning in      **3.56 hrs**

---

**Vehicle 02**      Returning in      **5.54 hrs**

---

**Vehicle 01** will be available first after      **0.72 hrs**

### STEP 06

Packages Remaining: 01

Vehicles Available: 01 | Current Time: 3.56 hrs

---

**PKG4**    ↗    **01 package**  
110kg

---

**Vehicle 01**      Delivering PKG4 (3.56+ 0.857)      **4.417 hrs**