```
import pandas as pd
import numpy as np
from math import ceil
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
# file location
path1 = '/content/drive/My Drive/cointab/Company X - Order Report.xlsx'
path2 = '/content/drive/My Drive/cointab/Company X - Pincode Zones.xlsx'
path3 = '/content/drive/My Drive/cointab/Company X - SKU Master.xlsx'
path4 = '/content/drive/My Drive/cointab/Courier Company - Invoice.xlsx'
path5 = '/content/drive/My Drive/cointab/Courier Company - Rates.xlsx'
```

```
df_Orders = pd.read_excel(path1)
df_pincodes = pd.read_excel(path2)
df_sku = pd.read_excel(path3)
df_courierInvoice = pd.read_excel(path4)
df_courierRates = pd.read_excel(path5)
```

```
# CALULATE TOTAL WEIGHT per SKU, COD PRICE
df_Orders_sum = df_Orders.merge(df_sku, on='SKU', how='left')
df_Orders_sum['Net Weight(kg)'] = (df_Orders_sum['Order Qty']*df_Orders_sum['Weight (g)']/1000).round(2)
df_Orders_sum['Net Price'] = df_Orders_sum['Item Price(Per Qty.)']

for i in range(df_Orders_sum.shape[0]):
  df_Orders_sum['Payment Mode'][i] = (0, 1)[ df_Orders_sum['Payment Mode'][i] == 'COD' ] ### mapping PREPAID-COD to 0-1 for calculative convi

for i in range(df_Orders_sum.shape[0]):
  if df_Orders_sum['Payment Mode'][i]:                                    ### if PAYMENT MODE = 1, i.e COD
    price = df_Orders_sum['Net Price'][i]
    if price > 300:
      df_Orders_sum['Net Price'][i] = 0.05*df_Orders_sum['Net Price'][i]
    else:
      df_Orders_sum['Net Price'][i] = 15
  else:                                                                   ### if PAYMENT MODE = 0, i.e PREPAID
    df_Orders_sum['Net Price'][i] = 0

df_Orders_sum.head()
```

```
<ipython-input-252-7cddd85fc583>:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable
  df_Orders_sum['Payment Mode'][i] = (0, 1)[ df_Orders_sum['Payment Mode'][i] == '
<ipython-input-252-7cddd85fc583>:15: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable
  df_Orders_sum['Net Price'][i] = 15
<ipython-input-252-7cddd85fc583>:13: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable
  df_Orders_sum['Net Price'][i] = 0.05*df_Orders_sum['Net Price'][i]
```

	ExternOrderNo	SKU	Order Qty	Payment Mode	Item Price(Per Qty.)	Weight (g)	Net Weight(kg)	Pr
0	2001827036	8904223818706	1	1	233	127	0.13	1
1	2001827036	8904223819093	1	1	233	150	0.15	1
2	2001827036	8904223819109	1	1	233	100	0.10	1
3	2001827036	8904223818430	1	1	470	165	0.16	2

```
#TOTAL_WEIGHT and COD_CHARGES for each ORDER_ID
### ORDER_ID is the KEY to both the mapping

total_weights = {}
COD_charges = {}

for i in range(df_Orders_sum.shape[0]):
```

```
  id = df_Orders_sum['ExternOrderNo'][i]
  total_weights[id] = total_weights.get(id, 0) + df_Orders_sum['Net Weight(kg)'][i]
  COD_charges[id] = COD_charges.get(id, 0) +df_Orders_sum['Net Price'][i]
```

```
##### Mapping PINCODES provided by COMPANY X to that of the COURIER COMPANY
pincode_map = {}
for i in range(df_pincodes.shape[0]):
  pincode_map[df_pincodes['Customer Pincode'][i]] = df_pincodes['Zone'][i]
print(len(pincode_map))

df_courierInvoice['Zone by X'] = df_courierInvoice['Customer Pincode'].map(pincode_map)
df_courierInvoice.head()
```

```
108
```

	AWB Code	Order ID	Charged Weight	Warehouse Pincode	Customer Pincode	Zone	Type of Shipment	Billing Amount (Rs.)	Zone by X
0	1091117222124	2001806232	1.30	121003	507101	d	Forward charges	140.0	d
1	1091117222194	2001806273	1.00	121003	486886	d	Forward charges	101.2	d
2	1091117222931	2001806408	2.50	121003	532484	d	Forward charges	224.6	d
3	1091117223244	2001806458	1.00	121003	143001	b	Forward charges	61.3	b
4	1091117229345	2001807012	0.15	121003	515591	d	Forward charges	45.4	d

```
### creating a Collective Order details consisting data from all the tables
df_order_details = pd.DataFrame()
df_order_details[['Order ID', 'AWB Number', 'Zone (Courier Company)', 'Zone (as per X)','Weight(courier comp.)', 'Billing Amount(Rs.)']] = df

df_order_details['Weight (as per X)'] = df_order_details['Order ID'].map(total_weights)
df_order_details['COD charge'] = df_order_details['Order ID'].map(COD_charges)

df_order_details.head()
```

	Order ID	AWB Number	Zone (Courier Company)	Zone (as per X)	Weight(courier comp.)	Billing Amount(Rs.)	Weight (as per X)	COD charge
0	2001806232	1091117222124	d	d	1.30	140.0	1.30	106.95
1	2001806273	1091117222194	d	d	1.00	101.2	0.62	88.25
2	2001806408	1091117222931	d	d	2.50	224.6	2.26	0.00
3	2001806458	1091117223244	b	b	1.00	61.3	0.70	0.00
4	2001807012	1091117229345	d	d	0.15	45.4	0.24	0.00

```
weight_slabs = {'a':0.25, 'b':0.50, 'c':0.75, 'd':1.25, 'e':1.50}
df_courierRates.set_index('Zone', inplace=True)
df_courierRates.head()
```

Zone	Weight Slabs	Forward Fixed Charge	Forward Additional Weight Slab Charge	RTO Fixed Charge	RTO Additional Weight Slab Charge
A	0.25	29.5	23.6	13.6	23.6
B	0.50	33.0	28.3	20.5	28.3
C	0.75	40.1	38.9	31.9	38.9
D	1.25	45.4	44.8	41.3	44.8
E	1.50	56.6	55.5	50.7	55.5

```
### calculate weight slabs for courier company
df_courier_slabs = df_order_details[['Order ID', 'Zone (Courier Company)', 'Weight(courier comp.)']]
df_courier_slabs['num_slabs'] = [0]*len(df_courier_slabs)
df_courier_slabs['weight_slabs_byCourier'] = [0]*len(df_courier_slabs)

for i in range(df_courier_slabs.shape[0]):
  num = ceil(df_courier_slabs['Weight(courier comp.)'][i] / weight_slabs[df_courier_slabs['Zone (Courier Company)'][i]])
  df_courier_slabs['num_slabs'][i] = num
  df_courier_slabs['weight_slabs_byCourier'][i] = num * weight_slabs[df_courier_slabs['Zone (Courier Company)'][i]]


df_courier_slabs.head()
```

```
<ipython-input-257-4d6e947d9027>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable
  df_courier_slabs['num_slabs'] = [0]*len(df_courier_slabs)
<ipython-input-257-4d6e947d9027>:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable
  df_courier_slabs['weight_slabs_byCourier'] = [0]*len(df_courier_slabs)
<ipython-input-257-4d6e947d9027>:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable
  df_courier_slabs['num_slabs'][i] = num
<ipython-input-257-4d6e947d9027>:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable
  df_courier_slabs['weight_slabs_byCourier'][i] = num * weight_slabs[df_courier_sl
```

	Order ID	Zone (Courier Company)	Weight(courier comp.)	num_slabs	weight_slabs_byCourier
0	2001806232	d	1.30	2	2.50
1	2001806273	d	1.00	1	1.25
2	2001806408	d	2.50	2	2.50
3	2001806458	b	1.00	2	1.00

```
df_expected_charge_calc = df_order_details[['Order ID', 'Zone (as per X)', 'Weight (as per X)', 'COD charge']]
df_expected_charge_calc = df_expected_charge_calc.merge(df_courierInvoice[['Order ID', 'Type of Shipment']], on='Order ID', how='left')

df_expected_charge_calc['num_slabs'] = [0]*len(df_expected_charge_calc)
df_expected_charge_calc['weight_slabs'] = [0]*len(df_expected_charge_calc)
for i in range(df_expected_charge_calc.shape[0]):
  num = ceil(df_expected_charge_calc['Weight (as per X)'][i] / weight_slabs[df_expected_charge_calc['Zone (as per X)'][i]])
  df_expected_charge_calc['num_slabs'][i] = num
  df_expected_charge_calc['weight_slabs'][i] = num * weight_slabs[df_expected_charge_calc['Zone (as per X)'][i]]

df_expected_charge_calc.head()
```

```
<ipython-input-258-7fd7b741a105>:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable
  df_expected_charge_calc['num_slabs'][i] = num
<ipython-input-258-7fd7b741a105>:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable
  df_expected_charge_calc['weight_slabs'][i] = num * weight_slabs[df_expected_char
```

	Order ID	Zone (as per X)	Weight (as per X)	COD charge	Type of Shipment	num_slabs	weight_slabs
0	2001806232	d	1.30	106.95	Forward charges	2	2.50
1	2001806273	d	0.62	88.25	Forward charges	1	1.25

```
###calculating total expected fare
forward_charge, rto_charge = [], []
for i in range(df_expected_charge_calc.shape[0]):
  id = df_expected_charge_calc['Zone (as per X)'][i].upper()
  extra_slabs = df_expected_charge_calc['num_slabs'][i]-1

  forward_charge.append(df_courierRates['Forward Fixed Charge'][id] + extra_slabs*df_courierRates['Forward Additional Weight Slab Charge'][id
  if df_expected_charge_calc['Type of Shipment'][i] == 'Forward charges':
    rto_charge.append(0)

  else:
    rto_charge.append(df_courierRates['RTO Fixed Charge'][id] + extra_slabs*df_courierRates['RTO Additional Weight Slab Charge'][id])
```

```
df_expected_charge_calc['forward_charge'] = forward_charge
df_expected_charge_calc['rto_charge'] = rto_charge
df_expected_charge_calc['total_sum'] = df_expected_charge_calc['COD charge'] + df_expected_charge_calc['forward_charge'] + df_expected_charge
df_expected_charge_calc.head(10)
```

	Order ID	Zone (as per X)	Weight (as per X)	COD charge	Type of Shipment	num_slabs	weight_slabs	forward_ch
0	2001806232	d	1.30	106.95	Forward charges	2	2.50	
1	2001806273	d	0.62	88.25	Forward charges	1	1.25	
2	2001806408	d	2.26	0.00	Forward charges	2	2.50	
3	2001806458	b	0.70	0.00	Forward charges	2	1.00	
4	2001807012	d	0.24	0.00	Forward charges	1	1.25	

```
df_order_details = df_order_details.merge(df_expected_charge_calc[['Order ID', 'weight_slabs', 'total_sum']], on='Order ID', how='left')
df_order_details.head()
```

	Order ID	AWB Number	Zone (Courier Company)	Zone (as per X)	Weight(courier comp.)	Billing Amount(Rs.)	Weight (as per X)
0	2001806232	1091117222124	d	d	1.30	140.0	1.30
1	2001806273	1091117222194	d	d	1.00	101.2	0.62
2	2001806408	1091117222931	d	d	2.50	224.6	2.26
3	2001806458	1091117223244	b	b	1.00	61.3	0.70

```
df_order_details = df_order_details.merge(df_courier_slabs[['Order ID', 'weight_slabs_byCourier']], on='Order ID', how='left')
df_order_details.head()
```

	Order ID	AWB Number	Zone (Courier Company)	Zone (as per X)	Weight(courier comp.)	Billing Amount(Rs.)	Weight (as per X)
0	2001806232	1091117222124	d	d	1.30	140.0	1.30
1	2001806273	1091117222194	d	d	1.00	101.2	0.62
2	2001806408	1091117222931	d	d	2.50	224.6	2.26
3	2001806458	1091117223244	b	b	1.00	61.3	0.70

```
df_order_details['Difference'] = df_order_details['total_sum']-df_order_details['Billing Amount(Rs.)']
df_order_details.head()
```

	Order ID	AWB Number	Zone (Courier Company)	Zone (as per X)	Weight(courier comp.)	Billing Amount(Rs.)	Weight (as per X)
0	2001806232	1091117222124	d	d	1.30	140.0	1.30
1	2001806273	1091117222194	d	d	1.00	101.2	0.62
2	2001806408	1091117222931	d	d	2.50	224.6	2.26
3	2001806458	1091117223244	b	b	1.00	61.3	0.70
4	2001807012	1091117229345	d	d	0.15	45.4	0.24

```
### REORDERING COLUMNS
df_order_details = df_order_details.loc[:,['Order ID', 'AWB Number', 'Weight (as per X)', 'weight_slabs', 'Weight(courier comp.)', 'weight_sl
                        'Zone (as per X)', 'Zone (Courier Company)', 'total_sum', 'Billing Amount(Rs.)', 'Difference']]
```

```
df_order_details.head()
```

	Order ID	AWB Number	Weight (as per X)	weight_slabs	Weight(courier comp.)	weight_slabs_b
0	2001806232	1091117222124	1.30	2.50	1.30	
1	2001806273	1091117222194	0.62	1.25	1.00	
2	2001806408	1091117222931	2.26	2.50	2.50	
3	2001806458	1091117223244	0.70	1.00	1.00	

```
### Preparing the SUMMARY DATAFRAME
count_corr, count_over, count_under = 0, 0, 0
amt_corr, amt_over, amt_under = 0, 0, 0
for i in range(df_order_details.shape[0]):
  if df_order_details['Difference'][i] == 0:
    count_corr += 1
    amt_corr += df_order_details['Billing Amount(Rs.)'][i]
  elif df_order_details['Difference'][i] > 0:
    count_under += 1
    amt_under += df_order_details['Difference'][i]
  elif df_order_details['Difference'][i] < 0:
    count_over += 1
    amt_over += df_order_details['Difference'][i]

df_summary = pd.DataFrame()
df_summary[''] = ['Total Orders - Correctly Charged', 'Total Orders - Over Charged', 'Total Order - Under Charged']
df_summary['Count'] = [count_corr, count_over, count_under]
df_summary['Amount'] = [amt_corr, amt_over, amt_under]

df_summary.head()
```

		Count	Amount
0	Total Orders - Correctly Charged	10	577.70
1	Total Orders - Over Charged	67	-4171.45
2	Total Order - Under Charged	47	4053.25

```
### RENAMING THE COLUMNS OF DATAFRAME
df_order_details.columns.values[:] = ['Order ID', 'AWB Number', 'Total weight as per X (KG)', 'Weight slab as per X (KG)', 'Total weight as p
df_order_details.head()
```

	Order ID	AWB Number	Total weight as per X (KG)	Weight slab as per X (KG)	Total weight as per Courier Company (KG)	Weight slab charged by Courier Company (KG)	Delivery Zone as per X	Delivery Zone charged by Courier Company	Expe Ch as p (
0	2001806232	1091117222124	1.30	2.50	1.30	2.50	d	d	19
1	2001806273	1091117222194	0.62	1.25	1.00	1.25	d	d	13
2	2001806408	1091117222931	2.26	2.50	2.50	2.50	d	d	9
3	2001806458	1091117223244	0.70	1.00	1.00	1.00	b	b	6

```
####saving order details to EXCEL files on drive
df_order_details.to_excel('/content/drive/My Drive/cointab/cointab.xlsx', index=False)
```

```
df_summary.to_excel('/content/drive/My Drive/cointab/summary.xlsx', index=False)
```