

LAB TEST – 2

TASK – 1:

CODE:

```
import io
```

```
def calculate_fee_averages(raw_text):
```

```
    """
```

Parses raw text to compute per-merchant averages of 'fee' and an overall average.

Args:

`raw_text (str)`: The raw text containing merchant data.

Returns:

`tuple`: A tuple containing:

- `dict`: A dictionary with merchant IDs as keys and their average fees as values.

- `float`: The overall average fee across all merchants.

```
    """
```

```
merchant_fees = {}
```

```
total_fee = 0
```

```
total_transactions = 0
```

```
# Use io.StringIO to treat the raw text as a file
```

```
data = io.StringIO(raw_text)
```

```
# Skip header line if present (assuming the first  
line is a header)
```

```
# You might need to adjust this based on your  
actual data format
```

```
# next(data)
```

```
for line in data:
```

```
    line = line.strip()
```

```
    if not line or line.startswith('#'): # Skip empty  
lines or comments
```

```
        continue
```

```
# Assuming data is comma-separated:
merchant_id, transaction_id, fee

try:

    merchant_id, transaction_id, fee_str =
line.split(',')

    fee = float(fee_str)

    if merchant_id not in merchant_fees:
        merchant_fees[merchant_id] = {'total_fee': 0,
'count': 0}

        merchant_fees[merchant_id]['total_fee'] += fee
        merchant_fees[merchant_id]['count'] += 1
        total_fee += fee
        total_transactions += 1
except ValueError:
    print(f"Skipping invalid line: {line}")
    continue

per_merchant_averages = {}
```

```
for merchant_id, data in merchant_fees.items():
    if data['count'] > 0:
        per_merchant_averages[merchant_id] =
data['total_fee'] / data['count']
    else:
        per_merchant_averages[merchant_id] = 0.0 #
Or handle as appropriate
```

```
overall_average = total_fee / total_transactions if
total_transactions > 0 else 0.0
```

```
return per_merchant_averages, overall_average
```

Sample raw text data

```
sample_raw_text = ""
```

```
merchant_A,txn_001,1.50
```

```
merchant_B,txn_002,2.00
```

```
merchant_A,txn_003,1.20
```

```
merchant_C,txn_004,3.00
```

```
merchant_B,txn_005,2.50
```

```
merchant_A,txn_006,1.80
```

```
.....
```

```
# Calculate averages
```

```
per_merchant_avg, overall_avg =  
calculate_fee_averages(sample_raw_text)
```

```
# Print results
```

```
print("Per-merchant averages:")
```

```
for merchant_id, avg_fee in  
per_merchant_avg.items():
```

```
    print(f"{merchant_id}: {avg_fee:.2f}")
```

```
print(f"\nOverall average fee: {overall_avg:.2f}")
```