```
# IMPORTANT: SOME KAGGLE DATA SOURCES ARE PRIVATE
# RUN THIS CELL IN ORDER TO IMPORT YOUR KAGGLE DATA SOURCES.
import kagglehub
kagglehub.login()
```

```
# IMPORTANT: RUN THIS CELL IN ORDER TO IMPORT YOUR KAGGLE DATA SOURCES,
# THEN FEEL FREE TO DELETE THIS CELL.
# NOTE: THIS NOTEBOOK ENVIRONMENT DIFFERS FROM KAGGLE'S PYTHON
# ENVIRONMENT SO THERE MAY BE MISSING LIBRARIES USED BY YOUR
# NOTEBOOK.

sajansinghshergill_apple_data_task_1_path = kagglehub.dataset_download('sajansinghshergill/apple-data-task-1')

print('Data source import complete.')
```

```
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as output when you create a versio
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session
```

```
/kaggle/input/apple-data-task-1/supermarket_transactions.xlsx
```

```
file_path = '/kaggle/input/apple-data-task-1/supermarket_transactions.xlsx'
data = pd.read_excel(file_path)

data.head()
```

	Unnamed: 0	id	timestamp	quantity	product_id	product_name	unit_price	total_amount	store	payment_me
0	0	47d54138-a950-4ec0-9d4a-e637e8dfb290	2022-04-28 09:11:00	10	56b274d1-b2a4-47be-abaf-6c1c7742beb9	wheat	0.39	3.90	South Billyview	
1	1	dea298b1-26ca-4a43-bcef-4050fb74ce1d	2019-12-06 16:21:00	1	56b274d1-b2a4-47be-abaf-6c1c7742beb9	wheat	0.39	0.39	South Edward	conta
2	2	8dcd78f6-2e68-4fe3-9bb0-afe3bb2c3944	2019-06-22 15:27:00	10	56b274d1-b2a4-47be-abaf-6c1c7742beb9	wheat	0.39	3.90	South Billyview	conta

```
# Filter for rows where product_name is "apples" and payment_method is "cash"
apples_cash = data[(data['product_name'] == 'apples') & (data['payment_method'] == 'cash')]

# Count the number of transactions for apples purchased with cash
apples_cash_count = apples_cash['quantity'].sum()
print("Number of apples purchased in cash:", apples_cash_count)
```

```
Number of apples purchased in cash: 0
```

```
# Sum the total_amount for apples purchased with cash
total_cash_on_apples = apples_cash['total_amount'].sum()
print("Total cash spent on apples:", total_cash_on_apples)
```

```
Total cash spent on apples: 0.0
```

```
# Filter for rows where store is "Bakershire" and customer_type is "non-member"
bakershire_non_member = data[(data['store'] == 'Bakershire') & (data['customer_type'] == 'non-member')]
```

```
# Sum the total_amount for these filtered rows
total_spent_bakershire_non_member = bakershire_non_member['total_amount'].sum()
print("Total money spent at Bakershire by non-member customers:", total_spent_bakershire_non_member)
```

```
Total money spent at Bakershire by non-member customers: 2857.5099999999993
```

```
# Task 1: Count of Apples Purchased with Cash
# Filter for rows where product_name is "apples" and payment_method is "cash"
apples_cash = data[(data['product_name'] == 'apples') & (data['payment_method'] == 'cash')]

# Count the number of apples purchased with cash
apples_cash_count = apples_cash['quantity'].sum()

# Task 2: Total Cash Spent on Apples
# Sum the total_amount for apples purchased with cash
total_cash_on_apples = apples_cash['total_amount'].sum()

# Task 3: Total Money Spent at Bakershire by Non-Member Customers Across All Payment Methods
# Filter for rows where store is "Bakershire" and customer_type is "non-member"
bakershire_non_member = data[(data['store'] == 'Bakershire') & (data['customer_type'] == 'non-member')]

# Sum the total_amount for these filtered rows
total_spent_bakershire_non_member = bakershire_non_member['total_amount'].sum()

apples_cash_count, total_cash_on_apples, total_spent_bakershire_non_member
```

```
(0, 0.0, 2857.5099999999993)
```