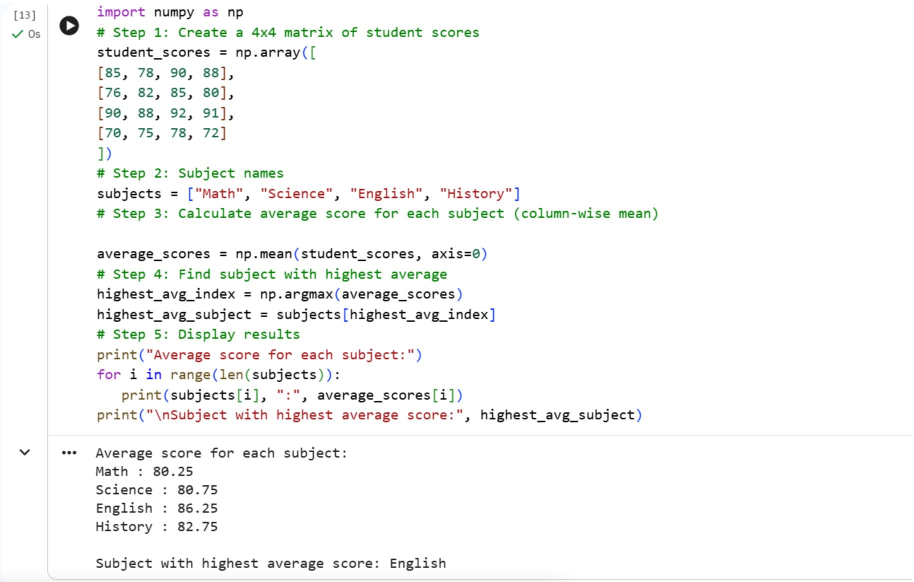
ROSENIVITHA J 192424009

DAY 1- 03/02/26

DSA0414 LAB EXP

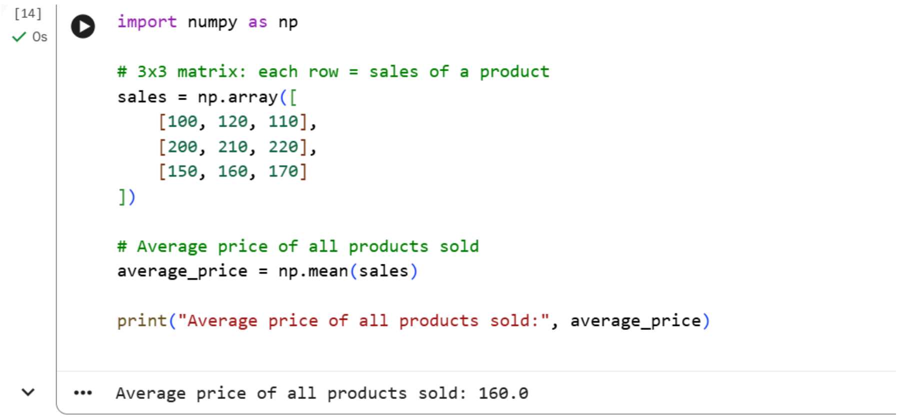
1



```python
import numpy as np
# Step 1: Create a 4x4 matrix of student scores
student_scores = np.array([
[85, 78, 90, 88],
[76, 82, 85, 80],
[90, 88, 92, 91],
[70, 75, 78, 72]
])
# Step 2: Subject names
subjects = ["Math", "Science", "English", "History"]
# Step 3: Calculate average score for each subject (column-wise mean)

average_scores = np.mean(student_scores, axis=0)
# Step 4: Find subject with highest average
highest_avg_index = np.argmax(average_scores)
highest_avg_subject = subjects[highest_avg_index]
# Step 5: Display results
print("Average score for each subject:")
for i in range(len(subjects)):
   print(subjects[i], ":", average_scores[i])
print("\nSubject with highest average score:", highest_avg_subject)
```

```
Average score for each subject:
Math : 80.25
Science : 80.75
English : 86.25
History : 82.75

Subject with highest average score: English
```

2

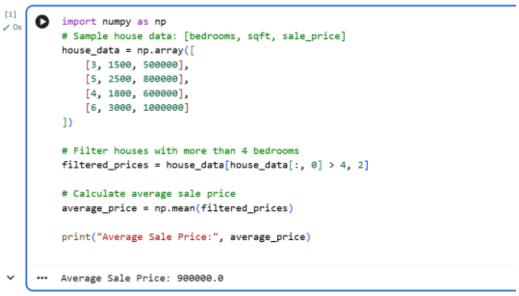


```python
import numpy as np

# 3x3 matrix: each row = sales of a product
sales = np.array([
    [100, 120, 110],
    [200, 210, 220],
    [150, 160, 170]
])

# Average price of all products sold
average_price = np.mean(sales)

print("Average price of all products sold:", average_price)
```

```
Average price of all products sold: 160.0
```

DAY 2 – 04/02/26

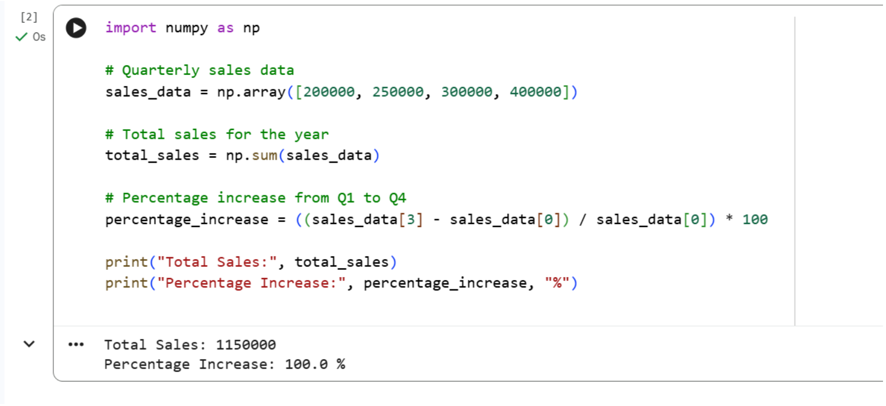
3



```
import numpy as np
# Sample house data: [bedrooms, sqft, sale_price]
house_data = np.array([
    [3, 1500, 500000],
    [5, 2500, 800000],
    [4, 1800, 600000],
    [6, 3000, 1000000]
])

# Filter houses with more than 4 bedrooms
filtered_prices = house_data[house_data[:, 0] > 4, 2]

# Calculate average sale price
average_price = np.mean(filtered_prices)

print("Average Sale Price:", average_price)
```

```
Average Sale Price: 900000.0
```

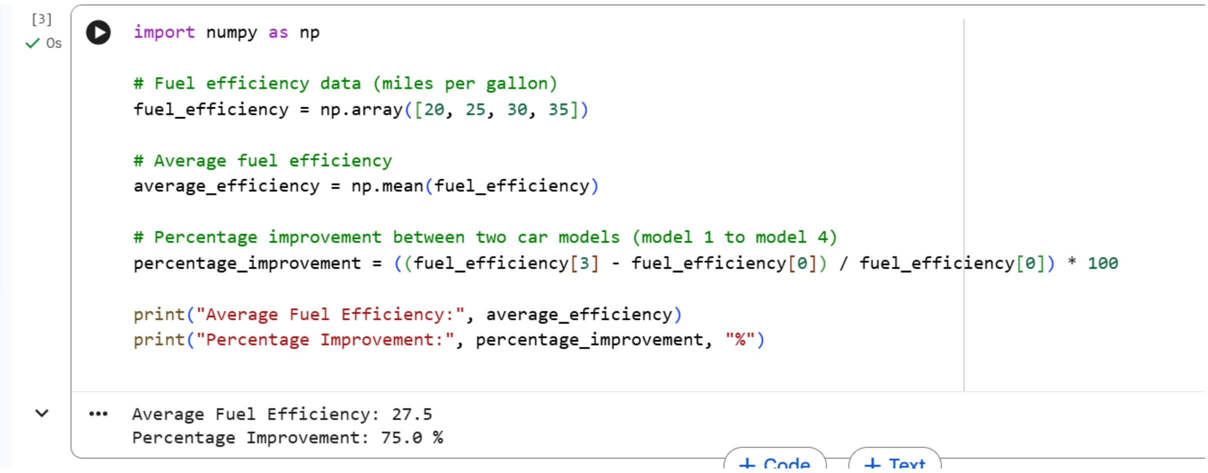
4



```
import numpy as np

# Quarterly sales data
sales_data = np.array([200000, 250000, 300000, 400000])

# Total sales for the year
total_sales = np.sum(sales_data)

# Percentage increase from Q1 to Q4
percentage_increase = ((sales_data[3] - sales_data[0]) / sales_data[0]) * 100

print("Total Sales:", total_sales)
print("Percentage Increase:", percentage_increase, "%")
```

```
Total Sales: 1150000
Percentage Increase: 100.0 %
```

5

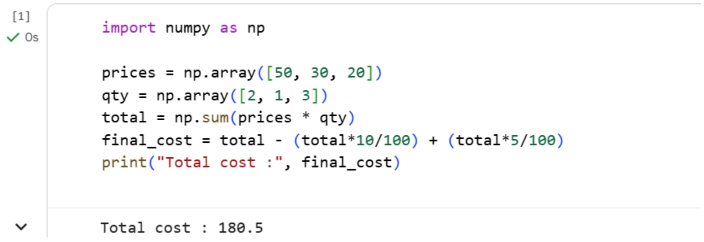


```
import numpy as np

# Fuel efficiency data (miles per gallon)
fuel_efficiency = np.array([20, 25, 30, 35])

# Average fuel efficiency
average_efficiency = np.mean(fuel_efficiency)

# Percentage improvement between two car models (model 1 to model 4)
percentage_improvement = ((fuel_efficiency[3] - fuel_efficiency[0]) / fuel_efficiency[0]) * 100

print("Average Fuel Efficiency:", average_efficiency)
print("Percentage Improvement:", percentage_improvement, "%")
```

```
Average Fuel Efficiency: 27.5
Percentage Improvement: 75.0 %
```

DAY 3- 05/02/26

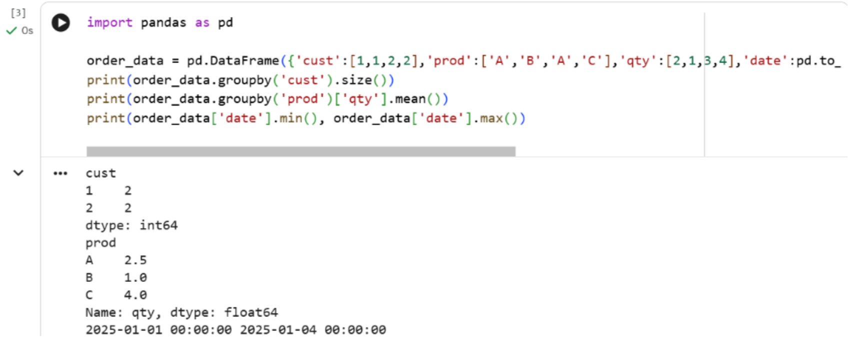
6



```
import numpy as np

prices = np.array([50, 30, 20])
qty = np.array([2, 1, 3])
total = np.sum(prices * qty)
final_cost = total - (total*10/100) + (total*5/100)
print("Total cost :", final_cost)
```

```
Total cost : 180.5
```

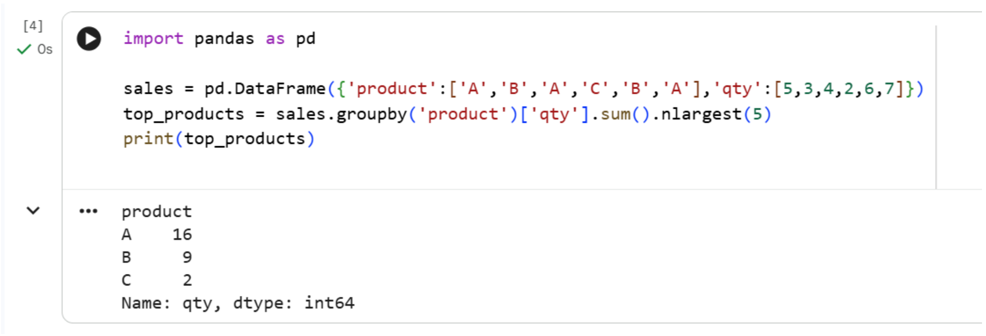
7



```
import pandas as pd

order_data = pd.DataFrame({'cust':[1,1,2,2],'prod':['A','B','A','C'],'qty':[2,1,3,4],'date':pd.to_
print(order_data.groupby('cust').size())
print(order_data.groupby('prod')['qty'].mean())
print(order_data['date'].min(), order_data['date'].max())
```

```
cust
1    2
2    2
dtype: int64
prod
A    2.5
B    1.0
C    4.0
Name: qty, dtype: float64
2025-01-01 00:00:00 2025-01-04 00:00:00
```

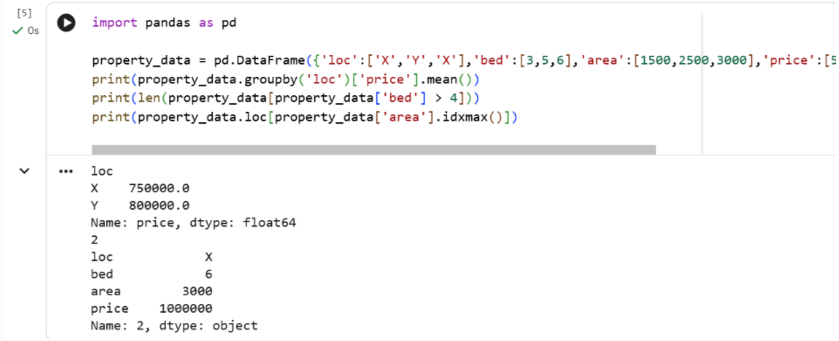
8



```
import pandas as pd

sales = pd.DataFrame({'product':['A','B','A','C','B','A'],'qty':[5,3,4,2,6,7]})
top_products = sales.groupby('product')['qty'].sum().nlargest(5)
print(top_products)
```

```
product
A    16
B     9
C     2
Name: qty, dtype: int64
```

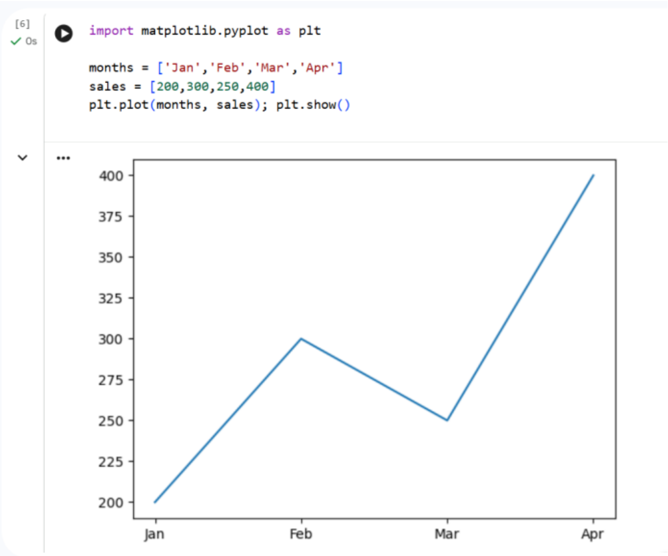
9



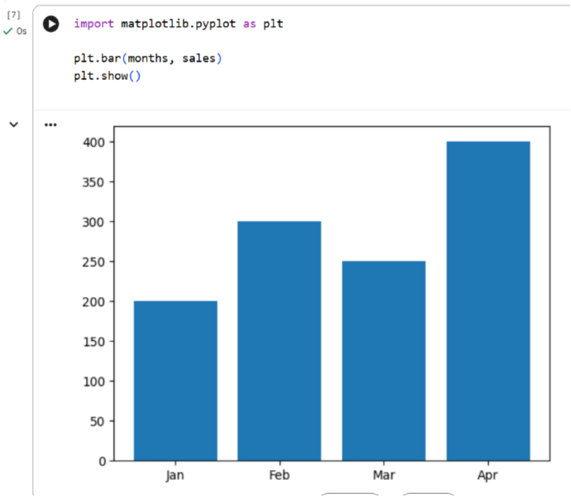
```
import pandas as pd

property_data = pd.DataFrame({'loc':['X','Y','X'],'bed':[3,5,6],'area':[1500,2500,3000],'price':[5
print(property_data.groupby('loc')['price'].mean())
print(len(property_data[property_data['bed'] > 4]))
print(property_data.loc[property_data['area'].idxmax()])
```

```
loc
X    750000.0
Y    800000.0
Name: price, dtype: float64
2
loc           X
bed           6
area       3000
price   1000000
Name: 2, dtype: object
```

10



```
import matplotlib.pyplot as plt

months = ['Jan','Feb','Mar','Apr']
sales = [200,300,250,400]
plt.plot(months, sales); plt.show()
```



```
import matplotlib.pyplot as plt

plt.bar(months, sales)
plt.show()
```