1.Student Performance Analysis (4×4 Matrix)

Program:-

```
import numpy as np
# 4x4 marks matrix: rows = students, columns = subjects
# Columns: [Math, Science, English, History]
student_scores = np.array([
    [85, 90, 78, 88],
    [92, 80, 85, 90],
    [75, 88, 82, 84],
    [89, 76, 91, 87]
])
subject_names = np.array(["Math", "Science", "English", "History"])
# Average score for each subject (column-wise mean)
subject_avgs = student_scores.mean(axis=0)
# Subject with highest average
max_index = subject_avgs.argmax()
highest_subject = subject_names[max_index]
print("Average score for each subject:")
for name, avg in zip(subject_names, subject_avgs):
    print(f"{name}: {avg:.2f}")
print("Subject with highest average score:", highest_subject)
```

Output:-

```
Average score for each subject:
Math: 85.25
Science: 83.50
English: 84.00
History: 87.25
Subject with highest average score: History
```

2.Sales Data Analysis (3×3 Matrix)

Program:-

```
import numpy as np
# 3x3 matrix: rows = products, columns = individual sale prices
sales_data = np.array([
    [200, 250, 220],
    [150, 180, 160],
    [300, 310, 290]
])
# Average price of all products sold
average_price = sales_data.mean()
print(f"Average price of all products sold in the past month: {average_price:.2f}")
```

Output:-

```
Average price of all products sold in the past month: 228.89
```

3. Housing Data (CSV Imported into NumPy)

Program:-

```
import numpy as np
# Columns: [bedrooms, square_footage, sale_price]
house_data = np.array([
    [3, 1200, 200000],
    [5, 1800, 300000],
    [4, 1500, 250000],
    [6, 2000, 350000],
    [5, 1700, 320000]
])
# Filter houses with more than 4 bedrooms
mask = house_data[:, 0] > 4
filtered_houses = house_data[mask]
# Extract sale prices from filtered houses (column index 2)
sale_prices = filtered_houses[:, 2]
# Average sale price of houses with more than 4 bedrooms
average_sale_price = sale_prices.mean()
print("Houses with more than 4 bedrooms:")
print(filtered_houses)
print(f"\nAverage sale price (more than 4 bedrooms): {average_sale_price:.2f}")
```

Output:-

```
Houses with more than 4 bedrooms:
[[     5   1800 300000]
 [     6   2000 350000]
 [     5   1700 320000]]
Average sale price (more than 4 bedrooms): 323333.33
```

4. Quarterly Sales Performance

Program:-

```
import numpy as np

# Quarterly sales data: [Q1, Q2, Q3, Q4]

sales_data = np.array([150000, 175000, 190000, 210000])

# Total sales for the year

total_sales = sales_data.sum()

# Percentage increase from Q1 to Q4

q1 = sales_data[0]

q4 = sales_data[3]

percentage_increase = ((q4 - q1) / q1) * 100

print(f"Total sales for the year: {total_sales}")

print(f"Percentage increase from Q1 to Q4: {percentage_increase:.2f}%")
```

Output:-

```
Total sales for the year: 725000

Percentage increase from Q1 to Q4: 40.00%
```

5. Fuel Efficiency Analysis

Program:-

```
import numpy as np

# Fuel efficiency of car models in miles per gallon (mpg)

fuel_efficiency = np.array([18.5, 20.0, 22.5, 25.0])

# Average fuel efficiency

average_efficiency = fuel_efficiency.mean()

# Compare model 0 and model 3

model1_index = 0

model2_index = 3

eff_model1 = fuel_efficiency[model1_index]

eff_model2 = fuel_efficiency[model2_index]

# Percentage improvement in fuel efficiency

percentage_improvement = ((eff_model2 - eff_model1) / eff_model1) * 100

print(f"Average fuel efficiency: {average_efficiency:.2f} mpg")

print(f"Fuel efficiency of Model {model1_index}: {eff_model1} mpg")

print(f"Fuel efficiency of Model {model2_index}: {eff_model2} mpg")

print(f"Percentage improvement from Model {model1_index} to Model {model2_index}:
{percentage_improvement:.2f}%")
```

Output:-

```
Average fuel efficiency: 21.50 mpg

Fuel efficiency of Model 0: 18.5 mpg

Fuel efficiency of Model 3: 25.0 mpg

Percentage improvement from Model 0 to Model 3: 35.14%
```