

```
In [1]: # Assignment (NumPy + Pandas Series)
import numpy as np

subjects = ["Math", "Science", "English", "History", "Computers"]

scores = np.array([
    [78, 85, 88, 70, 90],
    [60, 65, 70, 75, 80],
    [90, 88, 85, 92, 95],
    [55, 60, 58, 62, 65],
    [85, 82, 80, 78, 88],
    [72, 75, 78, 74, 76]
])

print("Scores array:")
print(scores)
print("\nSubjects order:", subjects)
```

```
Scores array:
[[78 85 88 70 90]
 [60 65 70 75 80]
 [90 88 85 92 95]
 [55 60 58 62 65]
 [85 82 80 78 88]
 [72 75 78 74 76]]
```

```
Subjects order: ['Math', 'Science', 'English', 'History', 'Computers']
```

```
In [10]: # A1.1 First 3 students
print("Task A1.1: First 3 students:")
print(scores[:3])
```

```
Task A1.1: First 3 students:
[[78 85 88 70 90]
 [60 65 70 75 80]
 [90 88 85 92 95]]
```

```
In [11]: # A1.2 ALL English scores (3rd subject)
print("Task A1.2: English scores:")
print(scores[:,2])
```

```
Task A1.2: English scores:
[88 70 85 58 80 78]
```

```
In [12]: # A1.3 Students with Computers>80
print("Task A1.3: Students with Computers > 80:")
mask_comp_gt80 = scores[:,4] > 80
print(mask_comp_gt80)
print(scores[mask_comp_gt80])
print("Computer scores:", scores[mask_comp_gt80,4])
```

Task A1.3: Students with Computers > 80:
 [True False True False True False]
 [[78 85 88 70 90]
 [90 88 85 92 95]
 [85 82 80 78 88]]
 Computer scores: [90 95 88]

```
In [13]: # A1.4 Students with avg > 75
print("Task A1.4: Students with avg > 75:")
student_averages = scores.mean(axis=1)
print("Student averages:", student_averages)
mask_avg_gt75 = student_averages > 75
print("Mask:", mask_avg_gt75)
print(scores[mask_avg_gt75])
```

Task A1.4: Students with avg > 75:
 Student averages: [82.2 70. 90. 60. 82.6 75.]
 Mask: [True False True False True False]
 [[78 85 88 70 90]
 [90 88 85 92 95]
 [85 82 80 78 88]]

```
In [7]: # A2 Score Summary
print("Task A2: Score summary (per subject):")
subj_avg = scores.mean(axis=0)
subj_max = scores.max(axis=0)
subj_min = scores.min(axis=0)
for i,s in enumerate(subjects):
    print(f"{s}: avg={subj_avg[i]:.2f}, max={subj_max[i]}, min={subj_min[i]}")
print("Highest overall average subject:", subjects[subj_avg.argmax()])
```

Task A2: Score summary (per subject):
 Math: avg=73.33, max=90, min=55
 Science: avg=75.83, max=88, min=60
 English: avg=76.50, max=88, min=58
 History: avg=75.17, max=92, min=62
 Computers: avg=82.33, max=95, min=65
 Highest overall average subject: Computers

```
In [14]: # Section B: Attendance
import pandas as pd

attendance = pd.Series([92,80,95,70,88,85], index=["S1","S2","S3","S4","S5","S6"])
print("Attendance:")
print(attendance)
```

Attendance:
 S1 92
 S2 80
 S3 95
 S4 70
 S5 88
 S6 85
 dtype: int64

```
In [15]: print("Task B1.1 Students with attendance < 85:")
print(attendance[attendance < 85])
```

Task B1.1 Students with attendance < 85:

S2 80

S4 70

dtype: int64

```
In [16]: print("Task B1.2 Count attendance >= 90:")
print((attendance >= 90).sum())
```

Task B1.2 Count attendance >= 90:

2

```
In [17]: print("Task B1.3 Average attendance:")
print(attendance.mean())
```

Task B1.3 Average attendance:

85.0

```
In [18]: print("Task B2.1 Compute average score per student from NumPy array:")
print(student_averages)
```

Task B2.1 Compute average score per student from NumPy array:

[82.2 70. 90. 60. 82.6 75.]

```
In [19]: print("Task B2.2 Created new Pandas Series with student labels S1..S6:")
avg_score_series = pd.Series(student_averages, index=["S1", "S2", "S3", "S4", "S5", "S6"])
print(avg_score_series)
```

Task B2.2 Created new Pandas Series with student labels S1..S6:

S1 82.2

S2 70.0

S3 90.0

S4 60.0

S5 82.6

S6 75.0

dtype: float64

```
In [20]: print("Task B2.3 Identify students with avg score > 75 AND attendance >= 85%:")
print("\nCondition mask:")
cond = (avg_score_series > 75) & (attendance >= 85)
print(cond)
print("Students satisfying both:", list(cond[cond].index))
```

Task B2.3 Identify students with avg score > 75 AND attendance >= 85%:

Condition mask:

S1 True

S2 False

S3 True

S4 False

S5 True

S6 False

dtype: bool

Students satisfying both: ['S1', 'S3', 'S5']

In []: