# IoT Sensor Readings Project  
# ==============================  
  
import array as arr # For Arrays  
# ------------------------------  
# 1. INTEGERS  
# ------------------------------  
sensor\_readings = [45, 67, 89, 34, 56, 78, 90, 23, 50]  
  
total = sum(sensor\_readings)  
average = total / len(sensor\_readings)  
minimum = min(sensor\_readings)  
maximum = max(sensor\_readings)  
  
print("=== INTEGERS ===")  
print("Total:", total)  
print("Average:", average)  
print("Minimum:", minimum)  
print("Maximum:", maximum)  
# ------------------------------  
# 2. STRINGS  
# ------------------------------  
report = f"IoT Sensor Report:\nTotal = {total}, Average = {average:.2f}\nMin = {minimum}, Max = {maximum}"  
print("\n=== STRINGS ===")  
print(report)  
# ------------------------------  
# 3. BOOLEANS  
# ------------------------------  
threshold = 60  
status = "Above Standard" if average > threshold else "Below Standard"  
compound\_status = average > threshold and maximum > 80  
  
print("\n=== BOOLEANS ===")  
print("Status:", status)  
print("Compound Status Condition Met:", compound\_status)  
# ------------------------------  
# 4. LISTS  
# ------------------------------  
print("\n=== LISTS ===")  
print("Original List:", sensor\_readings)  
# Add new reading  
sensor\_readings.append(72)  
# Remove readings below 30  
sensor\_readings = [x for x in sensor\_readings if x >= 30]  
# Sort readings  
sensor\_readings.sort()  
print("Modified & Sorted List:", sensor\_readings)  
# ------------------------------  
# 5. ARRAYS  
# ------------------------------  
sensor\_array = arr.array('i', [45, 67, 89, 34, 56, 78, 90, 23, 50])  
array\_sum = sum(sensor\_array)  
  
print("\n=== ARRAYS ===")  
print("Array Elements:", sensor\_array.tolist())  
print("Array Sum:", array\_sum)  
print("List Sum:", total)  
print("Comparison (Array vs List):", array\_sum == total)  
  
# ------------------------------  
# 6. DICTIONARIES  
# ------------------------------  
sensor\_records = [  
 {"id": 1, "name": "Sensor-A", "value": 45},  
 {"id": 2, "name": "Sensor-B", "value": 67},  
 {"id": 3, "name": "Sensor-C", "value": 89},  
]  
print("\n=== DICTIONARIES ===")  
print("Original Records:")  
for record in sensor\_records:  
 print(record)  
# Update one record  
sensor\_records[1]["value"] = 75 # Update Sensor-B  
# Delete another record  
del sensor\_records[0] # Remove Sensor-A  
  
# Add a new record  
sensor\_records.append({"id": 4, "name": "Sensor-D", "value": 55})  
# Compute total value across all records  
total\_values = sum(record["value"] for record in sensor\_records)  
  
print("\nUpdated Records:")  
for record in sensor\_records:  
 print(record)  
print("\nTotal Values Across Records:", total\_values)

INTEGERS  
# ------------------------------  
sensor\_readings = [45, 67, 89, 34, 56, 78, 90, 23, 50]  
  
total = sum(sensor\_readings)  
average = total / len(sensor\_readings)  
minimum = min(sensor\_readings)  
maximum = max(sensor\_readings)  
  
print("=== INTEGERS ===")  
print("Total:", total)  
print("Average:", average)  
print("Minimum:", minimum)  
print("Maximum:", maximum)  
# ------------------------------  
# 2. STRINGS  
# ------------------------------  
report = f"IoT Sensor Report:\nTotal = {total}, Average = {average:.2f}\nMin = {minimum}, Max = {maximum}"  
print("\n=== STRINGS ===")  
print(report)  
# ------------------------------  
# 3. BOOLEANS  
# ------------------------------  
threshold = 60  
status = "Above Standard" if average > threshold else "Below Standard"  
compound\_status = average > threshold and maximum > 80

print("\n=== BOOLEANS ===")  
print("Status:", status)  
print("Compound Status Condition Met:", compound\_status)  
# ------------------------------  
# 4. LISTS  
# ------------------------------  
print("\n=== LISTS ===")  
print("Original List:", sensor\_readings)  
# Add new reading  
sensor\_readings.append(72)  
# Remove readings below 30  
sensor\_readings = [x for x in sensor\_readings if x >= 30]  
# Sort readings  
sensor\_readings.sort()  
print("Modified & Sorted List:", sensor\_readings)  
# ------------------------------  
# 5. ARRAYS  
# ------------------------------  
sensor\_array = arr.array('i', [45, 67, 89, 34, 56, 78, 90, 23, 50])  
array\_sum = sum(sensor\_array)  
  
print("\n=== ARRAYS ===")  
print("Array Elements:", sensor\_array.tolist())  
print("Array Sum:", array\_sum)  
print("List Sum:", total)  
print("Comparison (Array vs List):", array\_sum == total)  
# ------------------------------  
# 6. DICTIONARIES  
# ------------------------------  
sensor\_records = [  
 {"id": 1, "name": "Sensor-A", "value": 45},  
 {"id": 2, "name": "Sensor-B", "value": 67},  
 {"id": 3, "name": "Sensor-C", "value": 89},  
]  
print("\n=== DICTIONARIES ===")  
print("Original Records:")  
for record in sensor\_records:  
 print(record)  
# Update one record  
sensor\_records[1]["value"] = 75 # Update Sensor-B  
# Delete another record  
del sensor\_records[0] # Remove Sensor-A  
  
# Add a new record  
sensor\_records.append({"id": 4, "name": "Sensor-D", "value": 55})  
  
# Compute total value across all records  
total\_values = sum(record["value"] for record in sensor\_records)  
  
print("\nUpdated Records:")  
for record in sensor\_records:  
 print(record)  
  
print("\nTotal Values Across Records:", total\_values)

descriptions and definitions about key words from my question are shown below:

**Main Keywords & Definitions**

* **Integer (int)** → Whole numbers (e.g., 45, 67) used for sensor values.
* **String (str)** → Text data (e.g., "Sensor-A", reports).
* **Boolean (bool)** → Logical values True or False used for threshold checks.
* **List** → An ordered, changeable collection (e.g., [45, 67, 89]) used for storing readings.
* **Array** → A data structure like a list, but more memory-efficient (from array module).
* **Dictionary (dict)** → Key-value pairs (e.g., {"id": 1, "name": "Sensor-A"}) used for sensor records.
* **Threshold** → A limit (e.g., 60) used to compare sensor averages.
* **Function (sum(), min(), max(), len())** → Built-in operations to compute totals, averages, and extremes.