Smart Factory Equipment Health Monitor

Scenario

A smart factory monitors the performance of machines using IoT sensors. Each machine sends temperature and vibration readings. The system must identify machines requiring maintenance, validate baseline machinery, and generate an equipment trace ID.

Requirements & Logic

1. Data Structures Setup

- **Dictionary (Machine_Readings):**
  Keys = Machine IDs (e.g., `'MCH-500'`)
  Values = Lists of readings (floats).
  Example:

  ```
  {'MCH-500': [72.5, 68.2], 'MCH-240': [88.0, 91.1],
  'MCH-105': [45.0, 49.5]}
  ```

- **Set (Baseline_Machines):**
  Contains IDs of critical machines that must always be active.
  Example:

  ```
  {'MCH-500', 'MCH-101', 'MCH-240'}
  ```

2. Filtering and Analysis (Loop/in/Update)

- Define a list `Maintenance_List` to store machine IDs that need inspection.
- Loop through the dictionary:
  - Check if the machine ID is part of `Baseline_Machines` — if yes, print a message like:
    `"MCH-500: BASELINE MACHINE — PRIORITY MONITORING"`.
  - Within each reading list, if any value is below 60.0, add that machine to the `Maintenance_List` and break further checks for that machine.

3. String Processing and Formatting

- Convert all Machine IDs into a tuple called `Machine_Tuple`.
- Join them with a hyphen (`-`) to form a single string `Health_Code`.
- Slice the string from the 4th to the 12th character.

- Convert the string to lowercase using `.lower()`.
- Use `.format()` to print:
    - The number of machines flagged for maintenance
    - The formatted health code string

Expected Output Example

```
MCH-500: BASELINE MACHINE — PRIORITY MONITORING
MCH-240: BASELINE MACHINE — PRIORITY MONITORING
Total Machines Requiring Maintenance: 1
Health Trace Code: -500-mch
Maintenance_List: ['MCH-105']
```