AIAC LAB TEST 2

SPORTS ANALYTICS TASKS

NAME: T. SAI VARUN

HT.NO:2403A510C0

BATCH NO: 05

TASK 1: FARE CALCULATOR

PROMPT TO AI

You are an expert Python developer.
Task: Implement a fare calculator for a sports analytics app.
Requirements:
- fare = km * base_per_km * surgeMultiplier.
- Surge applies strictly after 18:00 local time (HH:MM format).
- Exactly at 18:00 is NON-surge. E.g., 18:00 → no surge, 18:01 → surge.
- Round each fare to 2 decimals.
- Do not mutate input.
- No external libraries.
- Example:
  base_per_km = 18, surgeMultiplier = 2.0
  rides = [{'time': '08:00','km':3.0}, {'time':'18:30','km':5.0}]
  Output: [54.0,108.0]

PYTHON CODE

AIAC LAB TEST 2

ai.py > ...

```python
def calculate_fares(rides, base_per_km=18, surge_multiplier=2.0):
    """
    Calculate fare for each ride based on surge rules.
    Surge applies strictly after 18:00 (18:01 onwards).
    """
    fares = []
    for ride in rides:
        time_str = ride['time']
        km = ride['km']
        hour, minute = map(int, time_str.split(':'))

        # Check if surge applies
        if hour > 18 or (hour == 18 and minute > 0):
            fare = km * base_per_km * surge_multiplier
        else:
            fare = km * base_per_km

        fares.append(round(fare, 2))
    return fares


# -------- Quick Test --------
rides = [{'time': '08:00', 'km': 3.0},
         {'time': '18:00', 'km': 4.0},
         {'time': '18:01', 'km': 5.0},
         {'time': '19:15', 'km': 2.5}]
output = calculate_fares(rides)

print("Output:", output)
```

EXPECTED OUTPUT:

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS powershell

```
PS C:\Users\saiva\OneDrive\Desktop\lab> & C:\Users\saiva\AppData\Local\Programs\Python\Python313\python.exe c:/Users/saiva/OneDrive/
Desktop/lab/ai.py
Output: [54.0, 72.0, 180.0, 90.0]
PS C:\Users\saiva\OneDrive\Desktop\lab>
```

OBSERVATION

Parsing HH:MM correctly distinguishes surge times. Edge case 18:00 is non-surge, while 18:01 triggers surge. Fares are rounded to 2 decimals, and input is not mutated.

TASK 2: ROLLING KPI COMPUTATION BUG FIX

PROMPT TO AI

Fix the off-by-one bug in the rolling mean function for moving averages.

Requirements:

- All valid windows must be included: number of windows = len(xs) - w + 1.

AIAC LAB TEST 2

- Raise ValueError if w <= 0 or w > len(xs).

- Preserve simple O(n*w) approach.

- Example: xs = [9,10,11,12], w = 2 → [9.5,10.5,11.5].

Python Code:

AIAC LAB TEST 2

ai2.py > ...

```python
def rolling_mean(xs, w):
    """
    Compute rolling mean over xs with window size w.
    Includes all valid windows: total windows = len(xs) - w + 1.
    Guards invalid window sizes.
    """
    if w <= 0 or w > len(xs):
        raise ValueError("Invalid window size.")

    sums = []
    # FIX: iterate until len(xs) - w + 1 instead of len(xs) - w
    for i in range(len(xs) - w + 1):
        window = xs[i:i + w]
        sums.append(sum(window) / w)
    return sums

# --------- Tests ---------
def test_rolling_mean():
    # Failing test first: would miss the last window if bug exists
    xs = [9, 10, 11, 12]
    w = 2
    expected = [9.5, 10.5, 11.5]
    assert rolling_mean(xs, w) == expected, "Bug: Missing last window"

    # Edge cases
    assert rolling_mean([5], 1) == [5.0]
    try:
        rolling_mean([1, 2, 3], 0)
    except ValueError:
        pass
    else:
        raise AssertionError("Should raise ValueError for w <= 0")

    try:
        rolling_mean([1, 2, 3], 4)
    except ValueError:
        pass
    else:
        raise AssertionError("Should raise ValueError for w > len(xs)")

# Run tests and print result
try:
    test_rolling_mean()
    print("All tests passed.")
except AssertionError as e:
    print("Test failed:", e)

# Demo output for sample input
print("Sample Output:", rolling_mean([9, 10, 11, 12], 2))
```

AIAC LAB TEST 2

EXPECTED OUTPUT

```
PROBLEMS    OUTPUT    TERMINAL    DEBUG CONSOLE    PORTS                                   powershell

PS C:\Users\saiva\OneDrive\Desktop\lab> & C:\Users\saiva\AppData\Local\Programs\Python\Python313\python.exe c:/Users/saiva/OneDrive/Desktop/lab/ai2.py
All tests passed.
Sample Output: [9.5, 10.5, 11.5]
PS C:\Users\saiva\OneDrive\Desktop\lab>
```

OBSERVATION

The off-by-one bug was caused by iterating until len(xs)-w, which excluded the last window. Changing the loop to len(xs)-w+1 fixes the issue. Invalid window sizes are handled, and all tests pass. Complexity remains O(n*w).