Memory Allocation Simulator

Problem Description

You are simulating fixed-size contiguous memory partitioning. A system's memory is divided into equal-sized partitions. You are given a list of partition sizes (even though they're equal, they'll be given) and a list of process sizes that need to be allocated.

You must simulate First Fit, Best Fit, and Worst Fit allocation strategies, and calculate the total internal fragmentation after allocation.

Allocation Rules

- Each process must be allocated to exactly one partition.
- A process can only fit if its size ≤ partition size.
- Once a process is allocated, that partition is occupied and cannot be reused.
- If no partition fits a process, the process remains unallocated.
- Internal Fragmentation = (Sum of (partition size - process size) for all allocated processes).

Input Format

```
n                   // number of partitions
partition_sizes[n]     // sizes of each partition
m                   // number of processes
process_sizes[m]      // sizes of each process
```

Example Input

```
4
100 500 200 300
5
212 417 112 426 50
```

Output Format

```
StrategyName
Allocation: P1->Partition# / Not Allocated, ...
Total Internal Fragmentation: X
```

Example Output
First Fit
Allocation: P1->2 P2->4 P3->1 P4->Not Allocated P5->Not Allocated
Total Internal Fragmentation: 359

Best Fit
Allocation: P1->3 P2->4 P3->1 P4->Not Allocated P5->Not Allocated
Total Internal Fragmentation: 276

Worst Fit
Allocation: P1->2 P2->4 P3->1 P4->Not Allocated P5->Not Allocated
Total Internal Fragmentation: 359

Constraints

- $1 \leq n, m \leq 100$
- $1 \leq partition\_sizes[i], process\_sizes[i] \leq 10^4$

Task

Write a program to simulate all **three allocation strategies** and print results in the given format.
 You may implement it in **C, C++ or Java**.