# Railway Platform

Problem Submissions Discussions

**Time Limit: 2s, Memory Limit: 512MB**



### Problem Statement

On a railway platform, n passengers are standing in a single line. Each passenger carries a bag, and the weight of the bag carried by the i-th passenger is represented by an array a of length n, where $a_i$ denotes the bag weight of the passenger standing at position i. Two passengers at positions i and j decide to compare their bags. The comparison is valid if the following two conditiosn are satisfied -

- Passenger i must be standing ahead of passenger j, i.e., $i < j$
- And $a_i \geq a_j$

Your task is to determine the total number of valid comparisons possible among all such pairs.

### Input Format

- First line contains a single integer $t$ — the number of test cases.
- First line of each test case contains $n$ — the size of the array.
- The second line of each test case contains the array $a_1, a_2, a_3, \ldots, a_n$
- Summation of $n$ over all test cases doesn't exceed $2 \times 10^5$.

### Constraints

- $1 \leq t \leq 10^4$
- $2 \leq n \leq 2 \times 10^5$
- $1 \leq a_i \leq 10^9$

### Output Format

Find the total number of valid comparisons possible among all such pairs according to the problem statement.

### Sample Input 0

```
3
6
2 4 5 2 4 3
2
5 5
3
3 2 1
```

### Sample Output 0

```
8
1
3
```

### Explanation 0

In the first test case, valid comparisons are (considering 1-based indexing) -

- index (1,4) where $a_i >= a_j$ means $2 >= 2$
- index (2,4) where $a_i >= a_j$ means $4 >= 2$
- index (2,5) where $a_i >= a_j$ means $4 >= 4$
- index (2,6) where $a_i >= a_j$ means $4 >= 3$
- index (3,4) where $a_i >= a_j$ means $5 >= 2$
- index (3,5) where $a_i >= a_j$ means $5 >= 4$
- index (3,6) where $a_i >= a_j$ means $5 >= 3$
- index (5,6) where $a_i >= a_j$ means $4 >= 3$

C++

```cpp
#include <cmath>
#include <cstdio>
#include <vector>
#include <iostream>
#include <algorithm>
using namespace std;


int main() {
    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    return 0;
}

```

Line: 1 Col: 1

Upload Code as File  Test against custom input  Run Code  Submit Code