Sherlock is given an array of N integers ($A_0, A_1 \ldots A_{N-1}$ by Watson. Now Watson asks Sherlock how many different pairs of indices i and j exist such that i is not equal to j but A_i is equal to A_j.

That is, Sherlock has to count the total number of pairs of indices (i, j) where $A_i = A_j$ AND $i \neq j$.

Input Format

The first line contains T, the number of test cases. T test cases follow.

Each test case consists of two lines; the first line contains an integer N, the size of array, while the next line contains N space separated integers.

Output Format

For each test case, print the required answer on a different line.

Constraints

$1 \leq T \leq 10$

$1 \leq N \leq 10^5$

$1 \leq A[i] \leq 10^6$

Sample input

```
2
3
1 2 3
3
1 1 2
```

Sample output

```
0
2
```

Explanation

In the first test case, no two pair of indices exist which satisfy the given condition.

In the second test case as A[0] = A[1] = 1, the pairs of indices (0,1) and (1,0) satisfy the given condition.

```
def solve(a):
    myDict = {}
    for e in a:
        if e in myDict:
            myDict[e]+=1
        else:
            myDict[e]=1

    cnt=0
    for e in myDict:
        n=myDict[e]
        cnt+=n*(n-1)

    print(cnt)
    return cnt
```