

Pairwise AND sum

A bitwise AND takes two binary representations of equal length and performs the [logical AND](#) operation on each pair of corresponding bits. The result in each position is 1 if the first bit is 1 and the second bit is 1; otherwise, the result is 0. In this, we perform the multiplication of two bits; i.e., $1 \times 0 = 0$ and $1 \times 1 = 1$. For example:

```
0101 (decimal 5)
AND 0011 (decimal 3)
= 0001 (decimal 1)
```

You are given a sequence of **N** integer numbers **A**. Calculate the sum of **A_i AND A_j** for all the pairs **(i, j)** where **i < j**.

The **AND** operation is the **Bitwise AND** operation, defined first.

Input:

The first line of the input contains an integer **T** ($T \leq 10$) denoting the number of test cases. Each test case contains total number **N**. $N \leq 10^5$.

The second line contains **N** integer numbers - the sequence **A**. $A_i \leq 10^6$.

Output:

For each case, print the case number and find the best place to stand in the line so that you are selected.

Sample:

Input	Output
2	
5	
1 2 3 4 5	Case 1: 9
6	
1 2 3 4 5 6	Case 2: 21