Pitch Drop

Statement

Fluffy the Hamster is performing a pitch drop experiment, where a funnel is filled with a viscous (i.e. sticky/slowly-flowing) liquid, and left to slowly drip. He starts the experiment, and simultaneously opens the funnel and starts a stopwatch.

Every time a drop of liquid falls from the funnel, Fluffy records how many **seconds** have passed since the start of the experiment. The i-th droplet has timestamp T_i. In the end, he observes N droplets, and so he takes N timestamp measurements.

How many **matching pairs** of droplets are there which fell when the stopwatch read the same minute? In other words, if droplet A fell at 3h 20min 12s, and droplet B fell at 7h 20min 32s, they form a **matching pair**, as the stopwatch read the same minute (namely, 20min).

Given his data (all timestamps of droplets falling), how many such **matching pairs** of timestamps are there?

Constraints

- All timestamps are integers, representing the number of seconds since start of experiment.
- $2 \leq N \leq 10^5$
- $0 \leq T_i \leq 10^9$

NOTE: The input size is potentially large, usage of fast I/O routines is recommended.

Input

The first line of the input contains 1 integer N.

This is followed by N lines, describing each of the measurements. The i-th measurement line contains a single integer T_i, describing the number of seconds from the start of the experiment.

Output

Print the number of matching pairs.

Examples

Sample Input	Expected Output
6 34 3619 25919 4752021 694 1403	4

In the sample input, we have the conversion:

Number	Measurement (s)	HH:MM:SS
1	34	0h 0min 34s
2	3619	1h 0min 19s
3	25919	7h 11min 59s
4	4752021	1320h 0min 21s
5	694	0h 11min 34s
6	1403	0h 23min 23s

We then have the 4 matching pairs $(1, 2), (1, 4), (2, 4), (3, 5)$.

Notes

1. A skeleton file has been given to help you. You should not create a new file or rename the file provided. You should develop your program using this skeleton file.
2. You are free to define your own helper methods and classes (or remove existing ones) if it is suitable but you must put all the new classes, if any, in the same skeleton file provided.

Skeleton File

You are given the skeleton file `PitchDrop.java`. You should see the following contents when you open the file:

```
/**
 * Name          :
 * Matric. No    :
*/
```

```java
import java.util.*;

public class PitchDrop {
    private void run() {
    }

    public static void main(String args[]) {
        PitchDrop runner = new PitchDrop();
        runner.run();
    }
}
```