# Pair

You are organizing a competition, with N people wanting to take part in it. Each person has different strengths, i.e. no two people have the same strength.

To make the competition exciting, you want to pair two people in a team and pit them against each other. For this competition to be as exciting as possible, you want to match teams whose strengths are equal. You also want to have as many matches as possible by creating this rule: this competition is conducted in a "double round-robin" format, with a style similar to a "home-and-away" system. Furthermore, in a team, there is a team leader. Two matches / pairings are considered different if:

1. The members of the team are different.
2. The team leader (of a team) is different.
3. The "home" or "away" team is different.

For example, all these match pairings are considered different:

(A, B) vs (C, D)
(A, B) vs (C, E)
(B, A) vs (C, D)
(B, A) vs (D, C)

Note that in the pairings above, the team leader is listed first. Now, your task is to answer the following question: given a set of people competing with their own individual strengths, in how many different ways can you create the match pairings?

## Input

The first line of the input contains an integer N (N <= 1000), representing the number of people taking part in the competition. The next line contains N integers, the strength of each person. The $i^{th}$ integer denotes the strength of the $i^{th}$ person. No two people have equal strength.

## Output

Print the number of different pairings possible based on the strengths of the participants.

| Sample Input | Sample Output |
|---|---|
| 4<br>1 2 3 4 | 8 |

## Explanation

There are 8 possible pairings:

| | |
|---|---|
| (1, 4) vs (2, 3) | (2, 3) vs (1, 4) |
| (1, 4) vs (3, 2) | (2, 3) vs (4, 1) |
| (4, 1) vs (2, 3) | (3, 2) vs (1, 4) |
| (4, 1) vs (3, 2) | (3, 2) vs (4, 1) |

## Skeleton

You are given the skeleton file `Pair.java`.

## Notes

1. You are free to use anything to solve this problem.
2. To pass all test cases on CodeCrunch, your code needs to run in $O(N^2)$ or faster.