**Chemical Factory**

You are working in a chemical factory. Each chemical in the factory is represented by a string. You have to group the chemicals in such a way that no two explosive chemicals are together.

You are given an array of size N, denoting the names of N chemicals. Let *x* and *y* be two strings representing two chemicals.

Let *z = x + x (For example, if x=”abc” then z=”abcabc”)*.

The chemicals *x* and *y* are said to be explosive if we can jumble the characters of string *z* to get chemical *y*.

Your task is to return the **number of pairs** of chemicals that are explosive.

**Input Specification:**

|  |
| --- |
| **input1**: An integer N denoting the size of the array.  **input2**: An array of strings denoting the names of N chemicals. |

**Output Specification:**

|  |
| --- |
| An integer denoting the number of pairs of chemicals that are explosive. |

**Example 1:**

|  |
| --- |
| **input1**: 5  **input2**: {"hacker", "int", "niitnt", "hackhackerer", "long"}. |

|  |
| --- |
| **Output:**  2 |

**Explanation:**

There are two explosive pairs:

1. Pair {"int", "niitnt"} where x="int" and y="niitnt" is explosive as z = "intint" can be rearranged to form y="niitnt".​
2. Pair {"hacker", "hackhackerer"} where x="hacker" and y="hackhackerer" is explosive as z = "hackerhacker" can be rearranged​ to form y="hackhackerer". Therefore, the answer is 2.

**Example 2:**

|  |
| --- |
| **input1:** 3  **input2:** {“medium”, “easy”, “yeasseya”} |

|  |
| --- |
| **Output:**  1 |

**Explanation:**

There is only one explosive pair:

1. Pair {"easy", "yeasseya"} where x="easy" and y="yeasseya" is explosive as z = "easyeasy" can be rearranged to form y=​"yeasseya".​Therefore, the answer is 1.