You are given an array A of strings.

Two strings S and T are *special-equivalent* if after any number of *moves*, S == T.

A *move* consists of choosing two indices i and j with i % 2 == j % 2, and swapping S[i] with S[j].

Now, a *group of special-equivalent strings from A* is a non-empty subset S of A such that any string not in S is not special-equivalent with any string in S.

Return the number of groups of special-equivalent strings from A.

**Example 1:**

**Input:** ["a","b","c","a","c","c"]

**Output:** 3

**Explanation**: 3 groups ["a","a"], ["b"], ["c","c","c"]

**Example 2:**

**Input:** ["aa","bb","ab","ba"]

**Output:** 4

**Explanation**: 4 groups ["aa"], ["bb"], ["ab"], ["ba"]

**Example 3:**

**Input:** ["abc","acb","bac","bca","cab","cba"]

**Output:** 3

**Explanation**: 3 groups ["abc","cba"], ["acb","bca"], ["bac","cab"]

**Example 4:**

**Input:** ["abcd","cdab","adcb","cbad"]

**Output:** 1

**Explanation**: 1 group ["abcd","cdab","adcb","cbad"]

**Note:**

* 1 <= A.length <= 1000
* 1 <= A[i].length <= 20
* All A[i] have the same length.
* All A[i] consist of only lowercase letters.