Question 1 Every email consists of a local name and a domain name, separated by the @ sign. For example, in alice@gmail.com, alice is the local name, and gmail.com is the domain name. Besides lowercase letters, these emails may contain '.' or '+'.

If there are periods ('.') between some characters in the local name part of an email address, mail sent there will be forwarded to the same address without dots in the local name. For example, "alice.z@gmail.com" and "alicez@gmail.com" forward to the same email address. (Note that this rule does not apply for domain names.)

If there is a plus ('+') in the local name, everything after the first plus sign will be ignored. for example m.y+name@email.com will be forwarded to my@email.com. (Again, this rule does not apply for domain names.) It is possible to use both of these rules at the same time. Given a list of emails, we send one email to each address in the list. **How many different addresses actually receive mails**? Example:

Input:["test.email+alex@gmail.com","test.e.mail+bob@gmail.com","testemail+david@g.mail.com"]

Output: 2

Question 2 Given a string, you need to reverse the order of characters in each word within a sentence while still preserving whitespace and initial word order.

Example 1: Input: "Let's take this contest" Output: "s'teL ekat siht tsetnoc"

Question 3 Problem Description: You live in Grid City, which is composed of integer-numbered streets which run east-west (parallel to the x-axis) and integer-numbered avenues which run north-south (parallel to the y-axis).

The streets and avenues have infinite length, and there is a street for every integer y-coordinate and an avenue for every x-coordinate. All intersections are labelled by their integer coordinates: for example, avenue 7 and street -3 intersect at (7,-3).

You drive a special electric car which uses up one unit of electrical charge moving between adjacent intersections: that is, moving either north or south to the next street, or moving east or west to the next avenue). Until your battery runs out, at each intersection, your car can turn left, turn right, go straight through, or make a U-turn. You may visit the same intersection multiple times on the same trip.

Suppose you know your starting intersection, your destination intersection and the number of units of electrical charge in your battery. Determine whether you can travel from the starting intersection to the destination intersection using the charge available to you in such a way that your battery is empty when you reach your destination. For example: start is (10,2), end is (10,4), charge is 5. Output for this sample is N; start is (3,3), end is (4,7), charge is 9. Output for this sample is Y

Question 4 You are given an array A of strings. A *move onto S* consists of swapping any two even indexed characters of S, or any two odd indexed characters of S. Two strings S and T are *special-equivalent* if after any number of *moves onto S*, S == T.

For example, S = "zzxy" and T = "xyzz" are special-equivalent because we may make the moves "zzxy" -> "xzzy" -> "xyzz" that swap S[0] and S[2], then S[1] and S[3].

Now, a *group of special-equivalent strings from A* is a non-empty subset of A such that:

1. Every pair of strings in the group are special equivalent, and;
2. The group is the largest size possible (it is possible to only one string existed in a group)

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

Example 1:

Input: ["abcd","cdab","cbad","xyzz","zzxy","zzyx"]

Output: 3

Explanation:

One group is ["abcd", "cdab", "cbad"], since they are all pairwise special equivalent, and none of the other strings are all pairwise special equivalent to these.

The other two groups are ["xyzz", "zzxy"] and ["zzyx"]. Note that in particular, "zzxy" is not special equivalent to "zzyx".

Example 2:

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

Output: 3