# Problem 1:

**Pre-Implementaiton Notes:**

At first, this question reminded me of a similar question where you had to find all the connected friends in a social media platform. And essentially this question was to find the number of “friend groups” i guess you could say. I remember this was a graph related question, however, I immediatley was turned off by the idea for every letter in each word look for associated groups because that would take way too long.

Then I had an idea to associate each letter from a, b, c to a number 1, 2, 3.

And for each word you would combine all the seen letters in that word to the same letter

for example if you saw the word ‘abc’ appear now a, b, c, will all be associated to 1, 1, 1.

Next you run this algorithm for all words.

**Pre-Implementaiton Notes:**

This implementation above took way too long as well even with optimise data structures such as an `unordered_map` for the letter association to numbers, and then having all the numbers associated with the full words using an `unordered_multimap`.

So I remembered a conversation that had happened during my tutorial with a “union-find” strategy, and opting to this strategy allowed my code to pass all tests.