Given a string array words, return *an array of all characters that show up in all strings within the*words*(including duplicates)*. You may return the answer in **any order**.

**Example 1:**

**Input:** words = ["bella","label","roller"]

**Output:** ["e","l","l"]

**Example 2:**

**Input:** words = ["cool","lock","cook"]

**Output:** ["c","o"]

Step 1 : Find the count of occurrence of each character in a array

Step 2 : for i = 1 to n , for each time the loop is itreating and find the frequency of the current index and find the common chars between the exiting occurrence and current occurrence

Step 3 : After the final iteration , Calculate / add the chars to the result .

class Solution {

**// Return the frequency of the array**

    private int[] count(String s ){

        int len = s.length();

        int [] a = new int[26];

        for(int i = 0 ; i < len ; i++){

            a[s.charAt(i) - 'a']++;

        }

        return a;

    }

**//Count the common chars between two arrays**

private  int [] intersection(int []a , int [] b){

        for(int i = 0 ; i < 26 ; i++){

            a[i] = Math.min(a[i] , b[i]); **//Minimum occurrence would be the common chars**

        }

        return a;

    }

**//Printing the result**

    private List<String> result(int [] a){

        List<String> list = new ArrayList<>();

        for(int i = 0 ; i < a.length ; i++){

            if(a[i] != 0){

                String s = "";

                s += (char) ('a' + i);

                while(a[i] != 0){

                    list.add(s);

                    a[i]--;

                }

            }

        }

        return list;

    }

    public List<String> commonChars(String[] words) {

      int [] arr = count(words[0]); **// Getting the occurrence of the words[0]**

      int len = words.length;

      for(int i = 1 ; i < len ; i++){

        arr = intersection(arr , count(words[i])); **// Iteraring over all the other word in the array**

      }

      return result(arr);

    }

}