import java.io.File;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Comparator;
import java.util.List;
import java.util.Scanner;

public class ACSLStringDiff_KrishJain_J_2020 {
    public final static List<String> VOWELS = Arrays.asList("A", "E", "I", "O", "U");
    
    public static void main(String[] args) {
        for (int i = 0; i <= 5; i++) {
            Scanner s = new Scanner(System.in);
            String input[] = s.nextLine().split(" ");
            String firstString = input[0];
            String secondString = input[1];
            
            List<String> firstStringCharacters = Arrays.asList(firstString.split(""));
            List<String> secondStringCharacters = Arrays.asList(secondString.split(""));
            
            List<String> firstStringRemovedDoubles = removeDoubles(firstStringCharacters);
            List<String> secondStringRemovedDoubles = removeDoubles(secondStringCharacters);
            
            List<String> firstStringRemovedVowels = removeVowels(firstStringRemovedDoubles);
            List<String> secondStringRemovedVowels = removeVowels(secondStringRemovedDoubles);
            
            List<String> firstStringUniqueCharactersLToR = removeLikeCharactersAtSamePositionsLToR(
                firstStringRemovedVowels, secondStringRemovedVowels);
            List<String> secondStringUniqueCharactersLToR = removeLikeCharactersAtSamePositionsLToR(
                secondStringRemovedVowels, firstStringRemovedVowels);
            
            List<String> firstStringUniqueCharactersRToL = removeLikeCharactersAtSamePositionsRToL(
                firstStringUniqueCharactersLToR, secondStringUniqueCharactersLToR);
            List<String> secondStringUniqueCharactersRToL = removeLikeCharactersAtSamePositionsRToL(
                secondStringUniqueCharactersLToR, firstStringUniqueCharactersLToR);
            
            String processedFirstString = String.join("", firstStringUniqueCharactersRToL);
            String processedSecondString = String.join("", secondStringUniqueCharactersRToL);
            
            if (processedFirstString.length() == processedSecondString.length()) {
                System.out.println(processedFirstString.compareTo(processedSecondString) <= 0 ? processedFirstString : processedSecondString);
            } else {
                System.out.println(processedFirstString.length() < processedSecondString.length() ? processedFirstString : processedSecondString);
            }
        }
    }
}
private static List<String> removeDoubles(List<String> stringCharacters) {
    List<String> removedDoubles = new ArrayList<>();
    for (int i = 0; i < stringCharacters.size(); i++) {
        if (i == stringCharacters.size() - 1 ||
            !stringCharacters.get(i).equals(stringCharacters.get(i + 1))) {
            removedDoubles.add(stringCharacters.get(i));
        }
    }
    return removedDoubles;
}

private static List<String> removeVowels(List<String> stringCharacters) {
    List<String> stringCharactersWithoutFirstChar = stringCharacters.subList(1, stringCharacters.size());
    stringCharactersWithoutFirstChar.removeAll(VOWELS);
    stringCharactersWithoutFirstChar.add(0, stringCharacters.get(0));
    return stringCharactersWithoutFirstChar;
}

private static List<String> removeLikeCharactersAtSamePositionsLToR(List<String> removeFrom, List<String> comparingTo) {
    int shorterStringSize = Math.min(removeFrom.size(), comparingTo.size());
    List<String> stringUniqueCharacters = new ArrayList<>();
    for (int i = 0; i < shorterStringSize; i++) {
        if (!removeFrom.get(i).equals(comparingTo.get(i))) {
            stringUniqueCharacters.add(removeFrom.get(i));
        }
    }
    if (shorterStringSize < removeFrom.size()) {
        stringUniqueCharacters.addAll(removeFrom.subList(shorterStringSize, removeFrom.size()));
    }
    return stringUniqueCharacters;
}

private static List<String> removeLikeCharactersAtSamePositionsRToL(List<String> removeFrom, List<String> comparingTo) {
    int shorterStringSize = Math.min(removeFrom.size(), comparingTo.size());
    List<String> stringUniqueCharacters = new ArrayList<>();
    for (int i = 0; i < shorterStringSize; i++) {
        if (!removeFrom.get(removeFrom.size() - 1 - i).equals(comparingTo.get(comparingTo.size() - 1 - i))) {
            stringUniqueCharacters.add(0, removeFrom.get(removeFrom.size() - 1 - i));
        }
    }
    if (shorterStringSize < removeFrom.size()) {
        stringUniqueCharacters.addAll(0, removeFrom.subList(0, shorterStringSize));
    }
    return stringUniqueCharacters;
}