Pedro trabalha sempre até tarde todos os dias, com isso tem pouco tempo tempo para as tarefas domésticas. Para economizar tempo ele faz a lista de compras do supermercado em um aplicativo e costuma anotar cada item na mesma hora que percebe a falta dele em casa.

O problema é que o aplicativo não exclui itens duplicados, como Pedro anota o mesmo item mais de uma vez e a lista acaba ficando extensa. Sua tarefa é melhorar o aplicativo de notas desenvolvendo um código que exclua os itens duplicados da lista de compras e que os ordene alfabeticamente.

**Entrada**

A primeira linha de entrada contém um inteiro **N** (**N** < 100) com a quantidade de casos de teste que vem a seguir, ou melhor, a quantidade de listas de compras para organizar. Cada lista de compra consiste de uma única linha que contém de 1 a 1000 itens ou palavras compostas apenas de letras minúsculas (de 1 a 20 letras), sem acentos e separadas por um espaço.

**Saída**

A saída contém **N** linhas, cada uma representando uma lista de compra, sem os itens repetidos e em ordem alfabética.

| Exemplo de Entrada | Exemplo de Saída |
| --- | --- |
| 2 carne laranja suco picles laranja picles laranja pera laranja pera pera | carne laranja picles suco laranja pera |

***Solução***

import java.util.ArrayList;

import java.util.Iterator;

import java.util.List;

import java.util.Scanner;

import java.util.Set;

import java.util.TreeSet;

public class SupermarketShopping {

public static void main(String[] args) {

Scanner scr = new Scanner(System.in);

String produtos;

List<List<String>> listaProdutos = new ArrayList<>() {

private static final long serialVersionUID = 1L;

@Override

public String toString() {

StringBuilder stb = new StringBuilder();

for (int i = 0; i < this.size(); i++) {

stb.append(this.get(i) + " ");

}

return stb.toString();

}

};

Set<String> set = new TreeSet<>();

int N = scr.nextInt();

scr.nextLine();

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

produtos = scr.nextLine();

String[] produto = produtos.split(" ");

for (int j = 0; j < produto.length; j++) {

set.add(produto[j].toLowerCase());

}

List<String> listaP = new ArrayList<>() {

private static final long serialVersionUID = 1L;

@Override

public String toString() {

StringBuilder stb = new StringBuilder();

for (int i = 0; i < this.size(); i++) {

stb.append(this.get(i) + " ");

}

return stb.toString();

}

};

Iterator<String> iterator = set.iterator();

while (iterator.hasNext()) {

listaP.add(iterator.next());

}

listaProdutos.add(listaP);

set.clear();

}

listaProdutos.forEach(System.out::println);

scr.close();

}

}