

## H. Reunião Mágica

time limit per test: 2 seconds

memory limit per test: 256 megabytes

Um dos jogos de baralho mais famosos do mundo é o Reunião Mágica. Jogado por dois jogadores, cada um trás o seu próprio baralho de cartas para a partida e ganha quem conseguir derrotar o adversário. O que faz muito sucesso nesse jogo é que cada baralho é composto por monstros, magias e armadilhas dos mais diversos tipos, então os jogadores vivem montando diferentes baralhos para jogar.

Um dos modos de jogo do Reunião Mágica é o Unika, onde mesmo os jogadores tendo seu próprio baralho, não é permitido que ambos tenham alguma mesma carta em seus baralhos durante aquela partida. Então, antes de cada partida de Unika, os jogadores devem verificar as cartas de ambos para saber se seus baralhos estão aptos para a partida ou não.

Como essa tarefa demora muito tempo, você foi requisitado para criar um programa que identifique as cartas inválidas de um par de baralhos para uma partida de Unika.

### Input

A entrada consiste de múltiplos casos de testes. A primeira linha contém um inteiro  $T$  ( $1 \leq T \leq 10000$ ) que indica o número de casos de testes. As  $T$  linhas seguintes contêm duas strings  $R$  e  $S$  ( $1 \leq |R|, |S| \leq 1000$ ), separadas por um espaço, que são os baralhos de cada jogador, onde cada caracter, que pode ser uma letra ou um dígito, representa um carta.

### Output

A saída deve conter  $T$  linhas onde, para cada caso de teste, deve imprimir "Baralhos prontos para duelo" se não houver cartas que estejam presentes nos dois baralhos ao mesmo tempo, ou uma string, sem repetições e em ordem lexicográfica (alfabética), que são as cartas que aparecem nos dois baralhos ao mesmo tempo, conforme os exemplos.

### Examples

<b>input</b>
2
ABCDE FGHIJKL
abcde fghijkbl
<b>output</b>
Baralhos prontos para o duelo
b

<b>input</b>
2
123abc bca228
159abCA 149BADBBB9b
<b>output</b>
2abc
19Ab

[Copy](#) [Copy](#) [Copy](#) [Copy](#)**IDP - TAA - 2025/02****Private**

Participant

→ **About Group**

Este grupo tem o objetivo de organizar as atividades de programação da disciplina de Técnicas de Programação e Análise de Algoritmos.

[Group website](#)→ **Group Contests** ▼

- TAA - LEA 05
- TAA - LEE 05
- TAA - LEA 04
- TAA - LEE 04
- TAA - AS 01
- TAA - LEA 03
- TAA - LEE 03
- TAA - LEA 02
- TAA - LEE 02
- TAA - LEA 01
- TAA - LEE 01
- ET - Exercício de Testes

**TAA - LEE 04****Finished**

Contestant

→ **Last submissions**

Submission	Time	Verdict
<a href="#">344770587</a>	Oct/19/2025 22:37	Accepted

Supported by

