**Uncommon characters**

Submissions: [7323](https://practice.geeksforgeeks.org/problem_submissions.php?pid=2024)  Accuracy:

40.25%

   Difficulty: [Basic](https://practice.geeksforgeeks.org/Basic/0/0/)   Marks: 1

Associated Course(s): [Interview Preparation](https://practice.geeksforgeeks.org/courses/interview-preparation/)

Show Topic Tags   

[Amazon](https://practice.geeksforgeeks.org/company/Amazon/)

Find and print the **uncommon characters** of the two given strings **S1 and S2**. Here uncommon character means that either the character is present in one string or it is present in other string but **not in both**. The strings contains only **lowercase characters** and can contain **duplicates**.

**Input:**  
The first line of input contains an integer **T** denoting the number of test cases. Then T test cases follow. Each test case contains two strings in two subsequent lines.

**Output:**  
For each testcase, in a new line, print the uncommon characters of the two given strings in**sorted**order.

**Constraints:**  
1 <= T <= 100  
1 <= |S1|, |S2| <= 105

**Example:  
Input:**  
1  
characters  
alphabets  
**Output:**  
bclpr

\*\* For More Input/Output Examples Use ['Expected Output'](https://practice.geeksforgeeks.org/problems/uncommon-characters/0#ExpectOP) option \*\*

Contributor: Ayush Govil  
[Author: Ayush Govil 1](https://auth.geeksforgeeks.org/user/Ayush%20Govil%201/practice/)

<https://practice.geeksforgeeks.org/problems/uncommon-characters/0>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp2

{

class Program

{

static string UncommonCharacters(string a, string b)

{

HashSet<char> hash\_a = new HashSet<char>(a.ToCharArray());

HashSet<char> hash\_b = new HashSet<char>(b.ToCharArray());

SortedSet<char> sortedSet =

new SortedSet<char>();

foreach(char ch in hash\_a)

{

if(!hash\_b.Contains(ch))

{

sortedSet.Add(ch);

}

}

foreach (char ch in hash\_b)

{

if (!hash\_a.Contains(ch))

{

sortedSet.Add(ch);

}

}

return new string(sortedSet.ToArray());

}

static void Main(string[] args)

{

//string a = "characters";

//string b = "alphabets";

//Console.WriteLine(UncommonCharacters(a, b));

int t = int.Parse(Console.ReadLine());

while(t-- > 0)

{

string a = Console.ReadLine().Trim();

string b = Console.ReadLine().Trim();

Console.WriteLine(UncommonCharacters(a, b));

}

Console.ReadLine();

}

}

}