Jack and Daniel are friends. Both of them like letters, especially upper-case ones.
They are cutting upper-case letters from newspapers, and each one of them has his collection of letters stored in a stack.

One beautiful day, Morgan visited Jack and Daniel. He saw their collections. He wondered what is the lexicographically minimal string made of those two collections. He can take a letter from a collection only when it is on the top of the stack. Morgan wants to use all of the letters in their collections.

As an example, assume Jack has collected $a = [A, C, A]$ and Daniel has $b = [B, C, F]$. The example shows the top at index 0 for each stack of letters. Assembling the string would go as follows:

```
Jack    Daniel  result
ACA     BCF
CA      BCF     A
CA      CF      AB
A       CF      ABC
A       CF      ABCA
        F       ABCAC
                ABCACF
```

Note the choice when there was a tie at `CA` and `CF`.

Function Description

Complete the *morganAndString* function in the editor below. It should return the completed string.

morganAndString has the following parameter(s):

- *a*: a string representing Jack's letters, top at index 0
- *b*: a string representing Daniel's letters, top at index 0

Input Format

The first line contains the an integer t, the number of test cases.

The next t pairs of lines are as follows:
- The first line contains string a
- The second line contains string b.

Constraints

- $1 \leq T \leq 5$
- $1 \leq |a|, |b| \leq 10^5$
- a and b contain upper-case letters only, ascii[A-Z].

Output Format

Output the lexicographically minimal string $result$ for each test case in new line.

Sample Input

```
2
JACK
DANIEL
ABACABA
ABACABA
```

Sample Output

```
DAJACKNIEL
AABABACABACABA
```

Explanation

The first letters to choose from were J and D since they were at the top of the stack. D was chosen, the options then were J and A. A chosen. Then the two stacks have J and N, so J is chosen. (Current string is DAJ) Continuing this way till the end gives us the resulting string.