

# Morgan and a String

Jack and Daniel are friends. Both of them like letters, especially uppercase ones. They are cutting uppercase 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. Assemble the string 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.

*morganAndString* has the following parameter(s):

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

Returns

- *string*: the completed string

## 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]`.

## Sample Input

```
2
JACK
DANIEL
ABACABA
ABACABA
```

## Sample Output

```
DAJACKNIEL
AABABACABACABA
```

## Explanation

The first letters to choose from are J and D since they are at the top of the stack. D is chosen and the options now are J and A. A is chosen. Then the two stacks have J and N, so J is chosen. The current string is DA. Continue this way to the end to get the string.