

Problem A. Lex String

Time limit 1000 ms

Mem limit 262144 kB

Kuznecov likes art, poetry, and music. And strings consisting of lowercase English letters.

Recently, Kuznecov has found two strings, a and b , of lengths n and m respectively. They consist of lowercase English letters and **no character is contained in both strings**.

Let another string c be initially empty. Kuznecov can do the following two types of operations:

- Choose any character from the string a , remove it from a , and add it to the end of c .
- Choose any character from the string b , remove it from b , and add it to the end of c .

But, he can not do more than k operations of the same type in a row. He must perform operations until either a or b becomes empty. What is the lexicographically smallest possible value of c after he finishes?

A string x is lexicographically smaller than a string y if and only if one of the following holds:

- x is a prefix of y , but $x \neq y$;
- in the first position where x and y differ, the string x has a letter that appears earlier in the alphabet than the corresponding letter in y .

Input

There are several test cases in the input data. The first line contains a single integer t ($1 \leq t \leq 100$) — the number of test cases. This is followed by the test cases description.

The first line of each test case contains three integers n , m , and k ($1 \leq n, m, k \leq 100$) — parameters from the statement.

The second line of each test case contains the string a of length n .

The third line of each test case contains the string b of length m .

The strings contain only lowercase English letters. It is guaranteed that no symbol appears in a and b simultaneously.

Output

In each test case, output a single string c — the answer to the problem.

Examples

Input	Output
3 6 4 2 aaaaaa bbbb 5 9 3 caaca bedededeb 7 7 1 noskill wxhtzdy	aabaabaa aaabbcc dihktlwlxnyoz

Note

In the first test case, it is optimal to take two 'a's from the string a and add them to the string c . Then it is forbidden to take more characters from a , hence one character 'b' from the string b has to be taken. Following that logic, we end up with c being 'aabaabaa' when string a is emptied.

In the second test case it is optimal to take as many 'a's from string a as possible, then take as many 'b's as possible from string b . In the end, we take two 'c's from the string a emptying it.