Two Strings Game

Consider the following game for two players:

There are two strings A and B. Initially, some strings A' and B' are written on the sheet of paper. A' is always a substring of A and B' is always a substring of B. A move consists of appending a letter to **exactly one** of these strings: either to A' or to B'. After the move the constraint of A' being a substring of A and B' is a substring of B should still be satisfied. Players take their moves alternately. We call a pair (A', B') a position.

Two players are playing this game optimally. That means that if a player has a move that leads to his/her victory, he/she will definitely use this move. If a player is unable to make a move, he loses.

Alice and Bob are playing this game. Alice makes the first move. As always, she wants to win and this time she does a clever trick. She wants the starting position to be the K^{th} lexicographically winning position for the first player (i.e. her). Consider two positions (A'_1, B'_1) and (A'_2, B'_2). We consider the first position lexicographically smaller than the second if A1 is lexicographically smaller than A2, or if A1 is equal to A2 and B1 is lexicographically smaller than B2.

Please help her to find such a position, knowing the strings A, B and the integer *K*.

Input format
The first line of input consists of three integers, separated by a single space: N, M and K denoting the length of *A*, the length of *B* and K respectively. The second line consists of N small latin letters, corresponding to the string A. The third line consists of M small latin letters, corresponding to the string B.

Constraints
$1 <= N, M <= 3 * 10^5$
$1 <= K <= 10^{18}$

Output format
Output A' on the first line of input and B' on the second line of input. Please, pay attention that some of these strings can be empty. If there's no such pair, output "no solution" without quotes.

Sample input

```
2 2 5
ab
cd
```

Sample output

```
a
cd
```

Explanation of the example

```
Consider the position ("", "").
There are two different kinds of moves: either append the last letter to one of the strings, either to append the
first one. If the first player behaves in the first way, then second player can just do the same for another string
and win the game. So, the only chance that remains for the first player is to append the first letter to one of the
strings. Then, the second player can do the same. This way, after the first two moves we get ("a", "c"). Then
there's no option for the first player than to append the second letter to one of the strings. After this, there's
only one move that will be made by the second player. Then, we get ("ab", "cd"). This way, is the position ("", "")
is a losing one for the first player.

If we consider, for example, a position ("", "c"), the first player can make a move to make the position equal to
```

("a", "c"). After that, no matter what will the second player's move - ("ab", "c") or ("a", "cd"), the first will make the position ("ab", "cd") and the game will be ended with her victory. So, the position ("", "c") is a winning one for the first player.

The first five winning positions in the lexicographical order, starting with the first one are: ("", "c"), ("", "cd"), ("", "d"), ("a", ""), ("a", "cd").