Two Strings Game

HackerRank

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*.

Note: An empty string has higher precedence than character `"a"`

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 0

```
2 1 3
ab
c
```

Sample Output 0

```
a
```

c

Explanation 0

The given strings are ab and c. So there are $(2*2)*(2) = 8$ ways to fill a starting position (each character has two options, either to be present or not present).

$1.$ ["", ""] : If this is the start position, Alice will append a to $A^{'}$. So, the next two moves will consist of appending b and c to $A^{'}$ and $B^{'}$ respectively. So, Bob will suffer lack of moves and hence Alice wins.

$2.$ ["", "c"] : If this is the start position, Alice will append b to $A^{'}$. Now, Bob will suffer lack of moves and hence Alice wins.

$3.$ ["a", ""] : If Alice appends b to $A^{'}$ then Bob will append c to $B^{'}$ and if Alice appends c to $B^{'}$ then Bob will append b to $A^{'}$. So Alices looses.

$4.$ ["a", "c"] : If this is the start position, Alice will append b to $A^{'}$. Now, Bob will suffer lack of moves and hence Alice wins.

$5.$ ["ab", ""] : If this is the start position, Alice will append c to $B^{'}$. Now, Bob will suffer lack of moves and hence Alice wins.

$6.$ ["ab", "c"] : If this is the start position, Alice will suffer lack of moves and hence he looses.

$7.$ ["b", ""] : If this is the start position, Alice will append c to $B^{'}$. Now, Bob will suffer lack of moves and hence Alice wins.

$8.$ ["b", "c"] : If this is the start position, Alice will suffer lack of moves and hence he looses.

So, the list of start positions in lexicographical order where Alice wins are: ["", ""], ["", "c"], ["a", "c"], ["ab", ""], ["b", ""]. The 3^{rd} one in this list is ["a", "c"].