

Flip or Not

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 1024 megabytes

There are N cards arranged in a row. Initially, the i -th card from the left is face-up if the i -th character of the string S is 1, and face-down if it is 0. You can perform the following operation up to 10^6 times:

- Move the rightmost card to the leftmost position. If the moved card is face-up, then flip all the cards at positions A_1, A_2, \dots, A_P from the left. Additionally, you can choose to either flip all the cards at positions B_1, B_2, \dots, B_Q from the left or do nothing.

As a result of these operations, you want the i -th card from the left to be face-up if the i -th character of the string T is 1, and face-down if it is 0. Determine if it is possible to satisfy the condition with 10^6 or fewer operations, and if it is possible, output a sequence of operations that achieves the condition with the minimum number of operations.

Input

The input is provided in the following format from standard input:

```
 $N$   
 $S$   
 $T$   
 $P$   
 $A_1 A_2 \dots A_P$   
 $Q$   
 $B_1 B_2 \dots B_Q$ 
```

- All numerical inputs are integers.
- $1 \leq N \leq 5000$
- Each of S and T is a string of length N consisting of 0 and 1.
- $S \neq T$
- $1 \leq P, Q \leq N$
- $1 \leq A_1 < A_2 < \dots < A_P \leq N$
- $1 \leq B_1 < B_2 < \dots < B_Q \leq N$

Output

If it is impossible to satisfy the conditions with 10^6 or fewer operations, output -1. If it is possible to satisfy the conditions, output one operation sequence that minimizes the number of operations in the following format:

```
 $M$   
 $U$ 
```

Here, M is the number of operations, and U is a string of length M consisting only of 0 and 1 representing the operation sequence. If the i -th character of U is 1, it means that in the i -th operation, all cards from B_1, B_2, \dots, B_Q from the left are flipped. If the character is 0, no action is taken.

Examples

standard input	standard output
5 00001 00111 3 1 2 3 2 3 5	4 1001
4 0110 1000 2 1 2 4 1 2 3 4	-1

Note

For the first case, during the first operation, moving the rightmost card to the far left changes the card state to 10000. Since the moved card is facing up, we flip the cards at positions A_1 , A_2 , A_3 (1st, 2nd, 3rd) from the left, resulting in 01100. Next, if we choose to flip the front/back of the cards at positions B_1 , B_2 (3rd, 5th) from the left, the state becomes 01001. Continuing to follow the output example, in the second operation, the state changes to 01000, in the third operation to 00100, and in the fourth operation to 00111. There is no method to achieve this in fewer operations, so the output example is the correct output.

For the second test case, there is no way to satisfy the condition within 10^6 operations.