

Warehouse automation

The company is engaged in warehouse automation. The warehouse stores n types of goods, numbered from 1 to n , each type of product is stored in its own room. Product type i is stored in the room with the number i .

A special robot serves requests for receiving goods from the warehouse. To access the warehouse premises, the robot uses special electronic cards. Robot's cards are stored in a special compartment from which it can take out only the top card. The robot can return the removed card to the slot at any place: to the upper position, between any two cards, or to the lowest position.

To open the room, the robot acts as follows. He takes the cards out of the storage compartment and returns them back to the compartment until the card from the room that he needs to open is on the top position. After that, removing the card, the robot uses it to open the room, and then returns it to the card storage compartment. If in total the robot needed to remove x cards from the compartment, including the one with which he eventually opened the room, we will say that the robot performed x actions to open the room.

At the beginning of the working day, the robot received an order to issue m goods: a_1, a_2, \dots, a_m . Robot must deliver the goods exactly in that order. To do this, he consistently performs the following operations: opens the room in which the next product lies, takes the goods, closes the room and gives the goods to the client. After that, the robot proceeds to issue the next item.

The original electronic cards are in the following order, from top to bottom: b_1, b_2, \dots, b_n . For each room in the compartment there is exactly one card.

The time of goods delivery from the warehouse depends on how many times in total robot will have to remove the top card from the storage compartment to find the card from the next room. It is necessary in this way to choose the places where the robot must return the removed cards in order to minimize the total number of robot actions for opening the premises.

Write a program that given the integers n and m , the sequence of the issued goods a_1, a_2, \dots, a_m and the initial position of the cards in the storage compartment b_1, b_2, \dots, b_n determines the minimum number of actions the robot needs to perform in order to open all the rooms in the required order. For each card taken out, you must also indicate the position where it must be returned in order to achieve the optimal number of actions.

Input. First line contains two integers n and m ($1 \leq n, m \leq 3 * 10^5$) – the number of types of goods and the number of goods that must be issued from the warehouse.

Second line contains m integers a_1, a_2, \dots, a_m ($1 \leq a_i \leq n$) – the types of goods that need to be issued from the warehouse, listed in the order in which it is necessary to do.

Third line contains n different integers b_1, b_2, \dots, b_n ($1 \leq b_i \leq n$) – the order in which the cards are initially located in the storage compartment listed from top to bottom.

Output. The first line should contain the number k – the minimum number of actions that robot needs to perform in order to deliver all goods in specified order.

Then output k numbers. For each robot's action, print the position in the storage compartment where he must return the removed card. If the card returns to the top position, print 1, if after one card, print 2, and so on, for the last position you should print n .

If there are several ways to minimize the total number of actions, output any of them.

Explanation. In the second example, the cards in the robot compartment are moved as follows:

Действие	Перед действием	Извлеченная карта	Открытое помещение	Позиция, куда помещается карта	После действия
1	4, 3, 2, 1	4	4	4	3, 2, 1, 4
2	3, 2, 1, 4	3	–	4	2, 1, 4, 3
3	2, 1, 4, 3	2	–	2	1, 2, 4, 3
4	1, 2, 4, 3	1	1	4	2, 4, 3, 1
5	2, 4, 3, 1	2	2	4	4, 3, 1, 2
6	4, 3, 1, 2	4	4	1	4, 3, 1, 2
7	4, 3, 1, 2	4	4	4	3, 1, 2, 4

Sample input 1

1 1
1
1

Sample output 1

1
1

Sample input 2

4 5
4 1 2 4 4
4 3 2 1

Sample output 2

7
4 4 2 4 4 1 4

Sample input 3

2 2
1 2
2 1

Sample output 3

3
2 2 2