

## Автоматизация склада

Компания занимается автоматизацией склада. На складе хранятся  $n$  видов товаров, пронумерованных от 1 до  $n$ , каждый вид товара хранится в своём помещении. Товар вида  $i$  хранится в помещении с номером  $i$ .

Специальный робот обслуживает запросы по получению товаров со склада. Для доступа в помещения склада робот использует специальные электронные карты. Карты у робота хранятся в специальном отсеке, из которого он может вынуть верхнюю карту. Вынутую карту робот может вернуть в отсек на любое место: на верхнюю позицию, между любыми двумя картами или на самую нижнюю позицию.

Чтобы открыть помещение, робот действует следующим образом. Он вынимает карты из отсека для их хранения и возвращает их обратно в отсек, пока на верхней позиции не окажется карта от помещения, которое ему необходимо открыть. После этого, вынув эту карту, робот использует её, чтобы открыть помещение, и затем также возвращает в отсек для хранения карт. Если суммарно роботу потребовалось вынуть из отсека  $x$  карт, включая ту, которой он в итоге открыл помещение, будем говорить, что для открытия помещения робот совершил  $x$  действий.

В начале рабочего дня роботу поступил заказ на выдачу  $m$  товаров:  $a_1, a_2, \dots, a_m$ . Робот должен выдать товары именно в этом порядке. Для этого он последовательно выполняет следующие действия: открывает помещение, в котором лежит очередной товар, берет товар, закрывает помещение и выдаёт товар клиенту. После этого робот переходит к выдаче следующего товара.

Исходно электронные карты лежат в отсеке в следующем порядке, от верхней к нижней:  $b_1, b_2, \dots, b_n$ . Для каждого помещения в отсеке лежит ровно одна карта.

Время выдачи товаров со склада зависит от того, сколько раз суммарно роботу придётся вынимать верхнюю карту из отсека для их хранения, чтобы найти карту от очередного помещения. Необходимо таким образом выбрать места, куда робот должен возвращать вынутые карты, чтобы минимизировать суммарное количество действий робота для открытия помещений.

Напишите программу, которая по заданным целым числам  $n$  и  $m$ , последовательности выдаваемых товаров  $a_1, a_2, \dots, a_m$  и начальному положению карт в отсеке для хранения  $b_1, b_2, \dots, b_n$  определяет, какое минимальное количество действий придётся совершить роботу, чтобы открыть все помещения в необходимом порядке. Для каждой вынутой карты необходимо также указать позицию, на которую её необходимо вернуть, чтобы добиться оптимального количества действий.

**Вход.** Первая строка содержит два целых числа  $n$  и  $m$  ( $1 \leq n, m \leq 3 * 10^5$ ) – количество видов товаров и количество товаров, которые необходимо выдать со склада.

Вторая строка содержит  $m$  целых чисел  $a_1, a_2, \dots, a_m$  ( $1 \leq a_i \leq n$ ) – типы товаров, которые необходимо выдать со склада, перечисленные в том порядке, в котором это необходимо сделать.

Третья строка содержит  $n$  различных целых чисел  $b_1, b_2, \dots, b_n$  ( $1 \leq b_i \leq n$ ) – порядок, в котором карты исходно находятся в отсеке для их хранения, перечисленные от верхней к нижней.

**Выход.** Первая строка должна содержать число  $k$  – минимальное количество действий, которое потребуется совершить роботу, чтобы выдать товары в заданном порядке.

Далее выведите  $k$  чисел. Для каждого действия робота выведите одно число: позицию, на которую ему следует вернуть вынутую карту в отсек для хранения. Если карта возвращается на самую верхнюю позицию, следует вывести 1, если после одной карты, 2, и так далее, для последней позиции следует вывести  $n$ .

Если существует несколько способов минимизировать суммарное число действий, выведите любой из них.

**Пояснение.** Во втором примере карты в отсеке робота перемещаются следующим образом:

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

**Пример входа 1**

1 1  
1  
1

**Пример выхода 1**

1  
1

**Пример входа 2**

4 5  
4 1 2 4 4  
4 3 2 1

**Пример выхода 2**

7  
4 4 2 4 4 1 4

**Пример входа 3**

2 2  
1 2  
2 1

**Пример выхода 3**

3  
2 2 2