
Задача А. Кошелёк

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Рик превратил себя в кошелёк! Бум! Что вы на это скажете? Вот так поворот! Рик — кошелёк. Что вы на это скажете? Этого не происходило ещё никогда. Рик превратил себя в кошелёк! **КОШЕЛЁК-РИК!**

Так как Рик теперь не может двигаться, Морти вынужден помогать ему с обратным превращением. Для этого нужно правильным образом добавить в кошелёк несколько купюр. Всего есть n типов купюр, пронумерованных числами от 1 до n так, что чем больше номер купюры, тем больше ее номинал. В кошельке уже лежит некоторая сумма денег: для каждого i от 1 до n в кошельке уже лежит a_i купюр этого вида. Более того, купюры лежат там в ряд и отсортированы по номиналу, то есть если в кошельке лежат две купюры типов i и j , где $i < j$, то купюра типа i расположена левее купюры типа j . Это свойство должно сохраняться в каждый момент: по словам Рика, если его нарушить, последствия будут непредсказуемы!

Благо у Морти есть две изначально пустые стопки s_l и s_r , в которые он в процесс своих действий может класть купюры.

Кошелёк поддерживает 4 вида операций:

1. Вынуть из кошелька самую левую купюру и положить её с правого края стопки s_l .
2. Вынуть из кошелька самую правую купюру и положить её с левого края стопки s_r .
3. Вставить с левого края кошелька новую купюру.
4. Вставить с правого края кошелька новую купюру.

После каждой операции третьего или четвёртого типа необходимо тут же возвратить все вынутые купюры в кошелёк, а именно — все купюры стопки s_l , не меняя их расположения, поместить в левый край кошелька, а s_r — в правый край. В самом начале стопки s_l и s_r пустые.

Чтобы превратить Рика из кошелька назад в человека, Морти должен по очереди поместить в кошелёк m новых купюр, j -я добавленная купюра должна быть типа b_j . Купюры должны быть добавлены операциями 3-го и 4-го видов. После каждой операции купюры должны оставаться отсортированными по номиналу слева направо. Рик и Морти, чтобы понять, сколько времени у них займёт этот процесс, обратились к вам за помощью. Напишите программу, которая определит, какое наименьшее число операций первого и второго типов потребуется непосредственно перед каждым добавлением очередной купюры.

Формат входных данных

Первая строка входных данных содержит число n — количество различных типов купюр ($1 \leq n \leq 10^5$).

Вторая строчка содержит n целых чисел a_i — изначальное количество купюр каждого типа, которые лежат в кошельке ($0 \leq a_i \leq 10^5$).

В третьей строке находится число m — количество купюр, которые надо добавить в кошелёк, чтобы Рик обратно превратился в человека ($1 \leq m \leq 2 \cdot 10^5$).

В последней строке входных данных находится m чисел b_j — типы купюр, которые надо добавить в кошелёк, что бы Рик снова стал человеком ($1 \leq b_j \leq n$).

Формат выходных данных

В первой строке выходных данных выведите через пробел количество операций первого и второго типа, которое надо будет совершить перед каждым добавлением очередной купюры в кошелёк.

Пример

стандартный ввод	стандартный вывод
3 2 3 2 5 2 1 2 3 2	2 0 2 0 3

Замечание

Будем изображать текущее положение в формате $s_l/s/s_r$, где s — последовательность купюр, лежащих в кошельке.

В первом тесте в самом начале ситуация выглядит как $/1122233/$, после чего осуществляется 5 действий:

1. Чтобы добавить купюру типа 2, надо с любой из сторон дважды взять купюру, потом с той же стороны засунуть купюру типа 2.
2. В $/1122233/$ добавить купюру типа 1 можно, просто засунув эту купюру с левой стороны.
3. Из $/11122233/$ быстрее всего вынуть два раза купюру справа, получить $/111222/33$ и добавить справа купюру типа 2.
4. В $/111222233/$ сразу добавляем справа купюру типа 3.
5. Нет разницы, с какой стороны в $/1112222333/$ три раза вынуть купюру. Например, слева: получим $111/2222333/$ и добавим слева купюру типа 2, получив финальную ситуацию $/11122222333/$.