

# Очередь за кексами

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

В честь юбилея ректорат ЮФУ решил запустить акцию «Сто и десять кексов». В каждом корпусе университета открылась лавка с кексами, в которой каждый студент может получить бесплатные кексы.

Не прошло и пары минут после открытия, как к лавкам набежали студенты и образовалось много очередей. Но самая большая очередь образовалась в главном корпусе ЮФУ. Изначально в этой очереди стояло  $n$  студентов, но потом в течение следующих  $t$  минут какие-то студенты приходили и вставали в очередь, а какие-то уходили.

За каждым студентом закреплен номер его зачетной книжки, будем называть это число номером студента. У каждого студента будет уникальный номер, по которому можно однозначно его идентифицировать. Будем считать, что каждую минуту происходило одно из следующих событий:

1. Студент с номером  $x$  пришел и встал перед студентом с номером  $y$ ;
2. Студент с номером  $x$  пришел и встал в конец очереди;
3. Студент с номером  $x$  ушел из очереди; возможно, он потом вернется.

Аналитикам стало интересно, а какой будет очередь после  $t$  минут?

Помогите им и сообщите конечное состояние очереди.

## Формат входных данных

В первой строке заданы два целых числа  $n$  и  $t$  ( $1 \leq n, t \leq 10^5$ ) — текущее число студентов в очереди и количество изменений.

В следующей строке задается  $n$  целых **различных** чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ), где  $a_i$  — номер студента, который стоит на  $i$ -й позиции в очереди.

В следующих  $t$  строках идет описание запросов изменения очереди.

В каждой строке в зависимости от типа запроса задается два или три числа. Первое число  $t_j$  ( $1 \leq t_j \leq 3$ ) — тип события, которое произошло в  $j$ -ю минуту.

Если  $t_j = 1$ , то в строке задается еще 2 числа  $x$  ( $1 \leq x_j \leq 10^9$ ) и  $y$  ( $1 \leq y_j \leq 10^9$ ) — номер студента, который пришел, и номер студента, перед которым он встанет в очереди. Гарантируется, что студент с номером  $x$  ещё не занял очередь, а студент с номером  $y$  уже стоит в ней.

Если  $t_j = 2$ , то в строке задается еще 1 число  $x$  ( $1 \leq x_j \leq 10^9$ ) — номер студента, который пришел и встал в конец очереди. Гарантируется, что студент с номером  $x$  ещё не занял очередь.

Если  $t_j = 3$ , то в строке задается еще 1 число  $x$  ( $1 \leq x_j \leq 10^9$ ) — номер студента, который ушел из очереди. Гарантируется, что студент с номером  $x$  стоит в очереди.

## Формат выходных данных

В первой строке выведите одно число  $|a|$  — длину очереди после выполнения всех запросов изменения.

В следующей строке выведите  $|a|$  чисел  $a_1, a_2, \dots, a_{|a|}$ , где  $a_i$  — номер студента, который стоит на  $i$ -й позиции в очереди.

## Пример

стандартный ввод	стандартный вывод
7 6 1 2 3 4 5 6 7 1 8 3 2 9 3 3 1 3 9 2 10 3 1	9 2 8 4 5 6 7 3 9 10

## Замечание

Изначально очередь выглядит следующим образом:

1 2 3 4 5 6 7

В первую минуту приходит студент с номером 8 и встает перед студентом с номером 3.

1 2 (8) 3 4 5 6 7

Потом студент с номером 9 встает в конец очереди.

1 2 8 3 4 5 6 7 (9)

Студент с номером 3 уходит из очереди.

1 2 8 (3) 4 5 6 7 9

Потом он возвращается и становится перед студентом с номером 9.

1 2 8 4 5 6 7 (3) 9

После в конец очереди становится студент с номером 10.

1 2 8 4 5 6 7 3 9 (10)

И студент с номером 1 уходит из очереди.

(1) 2 8 4 5 6 7 3 9 10

После  $t$  событий очередь имеет следующий вид:

2 8 4 5 6 7 3 9 10