

## Задача А. Запрещенная сумма

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Вам дан массив целых чисел  $a$  длины  $n$ . Каждый элемент массива может быть «включенным» или «выключенным». Изначально все элементы включены.

Требуется добиться того, чтобы не существовало двух включенных элементов с суммой ровно  $k$ . То есть запрещена ситуация, в которой  $a_i$  и  $a_j$  оба включены,  $i \neq j$  и  $a_i + a_j = k$  для заранее известного  $k$ .

Определите, какое минимальное число элементов надо выключить, чтобы поставленное условие выполнялось.

### Формат входных данных

В первой строке даны два целых числа  $n$  и  $k$  ( $1 \leq n \leq 10^5$ ;  $1 \leq k \leq 10^9$ ).

В следующей строке перечислены  $n$  целых чисел  $a_i$  — элементы массива ( $1 \leq a_i \leq 10^9$ ).

### Формат выходных данных

Выведите единственное число — минимальное количество элементов, которое необходимо выключить.

### Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
0	–	примеры из условия		полная
1	20	$n \leq 20$	0	полная
2	15	$n, a_i, k \leq 1\,000$ ; $k$ нечетно		первая ошибка
3	25	$n, a_i, k \leq 1\,000$	0 – 2	первая ошибка
4	40	нет	0 – 3	первая ошибка

### Примеры

стандартный ввод	стандартный вывод
5 100 77 23 45 54 22	1
7 7 4 3 4 8 4 3 4	2

## Problem A. Forbidden Sum

Input file:            **standard input**  
Output file:          **standard output**  
Time limit:           **2 seconds**  
Memory limit:        **512 megabytes**

You are given an array of integers  $a$  of length  $n$ . Each element of the array can be “active” or “disabled”. Initially all elements are active.

It is required to ensure that there are no two active elements with a sum of exactly  $k$ . That is, the situation in which  $a_i$  and  $a_j$  are both active,  $i \neq j$  and  $a_i + a_j = k$  for a previously known  $k$  is forbidden.

Determine the minimum number of elements to be disabled so that the condition is satisfied.

### Input

The first line contains two integers  $n$  and  $k$  ( $1 \leq n \leq 10^5$ ;  $1 \leq k \leq 10^9$ ).

The next line lists  $n$  integers  $a_i$  — the elements of the array ( $1 \leq a_i \leq 10^9$ ).

### Output

Output a single number — the minimum number of elements to be disabled.

### Scoring

Points for each subtask will be awarded only if all tests for that subtask and the required subtasks have been successfully passed.

Subtask	Points	Additional Constraints	Required Subtasks	Feedback Policy
0	–	samples		full
1	20	$n \leq 20$	0	full
2	15	$n, a_i, k \leq 1000$ ; $k$ is odd		first error
3	25	$n, a_i, k \leq 1000$	0 – 2	first error
4	40	none	0 – 3	first error

### Examples

standard input	standard output
5 100 77 23 45 54 22	1
7 7 4 3 4 8 4 3 4	2

## Задача В. Рассадка по аудитории

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

В начале учебного года на урок программирования пришли  $k$  школьников. Места для посадки в аудитории разбиты на  $n$  рядов, пронумерованных от 1 до  $n$ . В каждом ряду  $m$  мест, пронумерованных от 1 до  $m$ . Поскольку это начало учебного года, необходимо в первую очередь определиться с рассадкой.

По различным причинам (по состоянию здоровья или просто потому что захотелось), у некоторых школьников есть ограничения по тому, где они могут сидеть. Ограничения бывают двух видов:

- школьник номер  $s_i$  может сидеть только на рядах с  $r_{i,1}$  по  $r_{i,2}$ ;
- школьник номер  $s_i$  может сидеть только на местах с номерами с  $c_{i,1}$  по  $c_{i,2}$ .

Всего таких ограничений ровно  $p$ . Для одного и того же школьника может быть не более одного ограничения каждого из двух видов. Назовем школьника *недовольным*, если ему не было выдано место в аудитории, либо если хотя бы одно из его ограничений не выполнено.

Изначально ни одному школьнику не было назначено место. Затем преподаватель  $q$  раз отдавал поручения вида «школьнику  $s_i$  сесть на место  $c_i$  в ряду  $r_i$ ». Если при этом

- или это место уже было кем-то занято,
- или школьник  $s_i$  уже сидел на каком-то другом месте, а при такой пересадке количество невыполненных ограничений **строго увеличивалось**,

поручение игнорировалось, и школьник  $s_i$  оставался на месте. Обратите внимание, что если у школьника не было назначенного ему места, а назначенное место свободно, поручение будет выполнено вне зависимости от ограничений школьника.

Обработайте все поручения преподавателя и выведите в конце количество недовольных школьников.

### Формат входных данных

В первой строке через пробел даны четыре целых числа  $n$ ,  $m$ ,  $k$  и  $p$  — размеры аудитории, количество школьников и количество ограничений ( $1 \leq n, m \leq 1000$ ,  $1 \leq k \leq 10^5$ ;  $0 \leq p \leq 2k$ ).

Каждая из следующих  $p$  строк содержит ограничение в формате «R  $s_i$   $r_{i,1}$   $r_{i,2}$ » или «C  $s_i$   $c_{i,1}$   $c_{i,2}$ » — либо ограничение на номер ряда, либо ограничение на номер места для школьника  $s_i$  ( $1 \leq s_i \leq k$ ;  $1 \leq r_{i,1} \leq r_{i,2} \leq n$ ;  $1 \leq c_{i,1} \leq c_{i,2} \leq m$ ). Гарантируется, что у каждого школьника есть не более одного ограничения каждого вида.

В следующей строке дано целое число  $q$  — количество поручений преподавателя ( $1 \leq q \leq 2 \cdot 10^5$ ). Каждая из следующих  $q$  строк описывает поручение преподавателя в формате «MOVE  $s_i$   $r_i$   $c_i$ » — школьнику  $s_i$  пересест в ряд  $r_i$  на место  $c_i$  ( $1 \leq s_i \leq k$ ;  $1 \leq r_i \leq n$ ;  $1 \leq c_i \leq m$ ).

### Формат выходных данных

Выведите единственное целое число — количество недовольных школьников в классе после обработки всех поручений преподавателя.

### Система оценки

В этой задаче всего 25 тестов, не считая примеров из условия. Каждый тест независимо от других оценивается в четыре балла. Тесты отличаются друг от друга размерами входных данных и различными дополнительными ограничениями.

## Примеры

стандартный ввод	стандартный вывод
2 2 4 0 3 MOVE 1 1 1 MOVE 2 1 1 MOVE 3 1 1	3
2 2 4 2 R 1 2 2 C 3 1 1 4 MOVE 1 1 1 MOVE 2 1 2 MOVE 3 2 1 MOVE 4 2 2	1
3 3 100 2 R 1 1 2 R 2 2 3 5 MOVE 1 2 3 MOVE 2 1 2 MOVE 100 1 1 MOVE 99 1 1 MOVE 98 2 2	97

## Problem B. Classroom Arrangement

Input file:            **standard input**  
Output file:           **standard output**  
Time limit:            2 seconds  
Memory limit:         256 megabytes

At the beginning of the school year,  $k$  students came to the programming lesson. The seats in the classroom are divided into  $n$  rows numbered from 1 to  $n$ . Each row has  $m$  seats numbered from 1 to  $m$ . Since this is the beginning of the school year, it is necessary to first decide on the seating.

For various reasons (health reasons or just because they want to), some students have restrictions on where they can sit. There are two types of restrictions:

- student number  $s_i$  can only sit in rows  $r_{i,1}$  through  $r_{i,2}$ ;
- student number  $s_i$  can only sit in seats numbered  $c_{i,1}$  through  $c_{i,2}$ .

In total, there are exactly  $p$  such constraints. For the same student there can be at most one constraint of each of the two types. Let's call a student *unhappy* if he was not given a place in the classroom or if at least one of his constraints is not satisfied.

Initially, no student was assigned a seat. Then the teacher gave  $q$  assignments of the form "student  $s_i$  moves to the seat  $c_i$  in the row  $r_i$ ". If at this time

- either this seat has already been occupied by someone,
- or student  $s_i$  **already sat in some other seat**, and with such a transfer the number of unfulfilled constraints **strictly increased**,

assignment was ignored, and student  $s_i$  remained in place. Note that if a student had no assigned seat prior and the assigned seat is free at the moment, this assignment will be completed regardless of the student's constraints.

Process all of the teacher's assignments and output the number of unhappy students at the end.

### Input

The first line contains four integers  $n$ ,  $m$ ,  $k$ , and  $p$  — the sizes of the classroom, the number of students, and the number of constraints ( $1 \leq n, m \leq 1000$ ;  $1 \leq k \leq 10^5$ ;  $0 \leq p \leq 2k$ ).

Each of the following  $p$  lines contains a constraint in the format "**R**  $s_i$   $r_{i,1}$   $r_{i,2}$ " or "**C**  $s_i$   $c_{i,1}$   $c_{i,2}$ " — either a row number constraint or a seat number constraint for student  $s_i$  ( $1 \leq s_i \leq k$ ;  $1 \leq r_{i,1} \leq r_{i,2} \leq n$ ;  $1 \leq c_{i,1} \leq c_{i,2} \leq m$ ). It is guaranteed that each student has at most one constraint of each type.

The next line contains an integer  $q$  — the number of teacher's assignments ( $1 \leq q \leq 2 \cdot 10^5$ ). Each of the following  $q$  lines describes a teacher's assignment in the format "**MOVE**  $s_i$   $r_i$   $c_i$ " — the student  $s_i$  has to sit in row  $r_i$  on seat  $c_i$  ( $1 \leq s_i \leq k$ ;  $1 \leq r_i \leq n$ ;  $1 \leq c_i \leq m$ ).

### Output

Print a single integer — the number of unhappy students in the class after processing all of the teacher's assignments.

### Scoring

There are 25 tests in this problem in total excluding the samples from the statement. Each test is independently graded with four points. The tests differ from each other by the size of the input data and various additional constraints.

## Examples

standard input	standard output
2 2 4 0 3 MOVE 1 1 1 MOVE 2 1 1 MOVE 3 1 1	3
2 2 4 2 R 1 2 2 C 3 1 1 4 MOVE 1 1 1 MOVE 2 1 2 MOVE 3 2 1 MOVE 4 2 2	1
3 3 100 2 R 1 1 2 R 2 2 3 5 MOVE 1 2 3 MOVE 2 1 2 MOVE 100 1 1 MOVE 99 1 1 MOVE 98 2 2	97

## Задача С. Неслучайные блуждания

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

Перед вами в ряд расположены  $n$  книг,  $i$ -я книга в ряду имеет *познавательность*, равную  $a_i$ . Вы изучаете книги следующим образом:

1. сперва выбирается какая-то книга, обозначим ее номер за  $x$ ;
2. затем вы начинаете перебирать книги подряд, начиная с нее, по направлению влево (направлению уменьшения номера);
3. пока слева от текущей книги находится книга меньшей или равной познавательности, вы перемещаетесь к ней;
4. при этом вам нравится разнообразие, поэтому вы не сделаете больше  $k$  перемещений между книгами с одинаковой познавательностью.

Иными словами, вы движетесь влево, пока не дойдете до книги с большей весомостью, начала массива или  $(k + 1)$ -й книги подряд с одинаковой познавательностью.

Например, если познавательности книг равны  $\langle 3, 3, 3, 4, 4, 5 \rangle$ ,  $k = 2$ , и вы начинаете с **последней** книги, вы совершите четыре перемещения влево, после чего остановитесь.

Для каждой из  $m$  возможных «стартовых» книг  $x_i$  определите, на какой книге остановится такой процесс.

### Формат входных данных

В первой строке дано целое число  $n$  — количество книг ( $1 \leq n \leq 4 \cdot 10^5$ ). Во второй строке через пробел перечислены  $n$  целых чисел  $a_i$  — значения познавательности книг в порядке их следования в ряду ( $1 \leq a_i \leq 10^9$ ).

В следующей строке через пробел даны два целых числа  $m$  и  $k$  — количество запросов и максимальное число перемещений между книгами равной познавательности ( $1 \leq m \leq 4 \cdot 10^5$ ;  $0 \leq k \leq n$ ).

В последней строке через пробел перечислены  $m$  целых чисел  $x_i$  — номера книг, с которых будет начинаться описанный процесс ( $1 \leq x_i \leq n$ ).

### Формат выходных данных

Выведите через пробел  $m$  целых чисел от 1 до  $n$  — номера книг, на которых остановится процесс в каждом из запросов.

### Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
0	–	примеры из условия		полная
1	13	$n, m \leq 10$	0	полная
2	12	$n, m \leq 1000, k = 0$		полная
3	15	$n, m \leq 1000, k = 1$		полная
4	18	$n, m \leq 1000$	0 – 3	первая ошибка
5	19	$k = 0$	2	первая ошибка
6	23	нет	0 – 5	первая ошибка

## Примеры

стандартный ввод	стандартный вывод
6 3 3 3 4 4 5 4 2 3 4 5 6	1 1 2 2
7 1 5 7 2 10 10 6 7 0 1 2 3 4 5 6 7	1 1 1 4 4 6 7



## Problem C. Non-Random Walk

Input file:            **standard input**  
Output file:           **standard output**  
Time limit:            **2 seconds**  
Memory limit:         **256 megabytes**

There are  $n$  books in a row in front of you, the  $i$ -th book in the row has *popularity* equal to  $a_i$ .

You study the books as follows:

1. first you choose some book, let's denote its number by  $x$ ;
2. then you start going through the books in a row, starting from it, to the left direction (the direction of decreasing number);
3. as long as there is a book of lesser or equal popularity value to the left of the current book, you move to it;
4. you like variety, so you don't make more than  $k$  moves between books with the same popularity.

In other words, you move to the left until you reach the book with higher popularity, the beginning of the array, or the  $(k + 1)$ -th book in a row with the same popularity value.

For example, if the popularities of the books are  $\langle 3, 3, 3, 4, 4, 5 \rangle$ ,  $k = 2$ , and you start at the **last** book, you will make four moves to the left and then stop.

For each of  $m$  possible "starting" books  $x_i$  determine at which book such a process will stop.

### Input

The first line contains an integer  $n$  — the number of books ( $1 \leq n \leq 4 \cdot 10^5$ ). The second line lists  $n$  integers  $a_i$  — popularity values of the books in their order in a row ( $1 \leq a_i \leq 10^9$ ).

In the next line two integers  $m$  and  $k$  representing the number of queries and the maximum number of moves between books of equal popularities are given ( $1 \leq m \leq 4 \cdot 10^5$ ;  $0 \leq k \leq n$ ).

The last line lists  $m$  integers  $x_i$  — the numbers of books from which the process will start ( $1 \leq x_i \leq n$ ).

### Output

Print  $m$  space-separated integers from 1 to  $n$  — the numbers of books the process will stop at in each query.

### Scoring

Points for each subtask will be awarded only if all tests for that subtask and the required subtasks have been successfully passed.

Subtask	Points	Additional Constraints	Required Subtasks	Feedback Policy
0	–	samples		full
1	13	$n, m \leq 10$	0	full
2	12	$n, m \leq 1000, k = 0$		full
3	15	$n, m \leq 1000, k = 1$		full
4	18	$n, m \leq 1000$	0 – 3	first error
5	19	$k = 0$	2	first error
6	23	none	0 – 5	first error

## Examples

standard input	standard output
6 3 3 3 4 4 5 4 2 3 4 5 6	1 1 2 2
7 1 5 7 2 10 10 6 7 0 1 2 3 4 5 6 7	1 1 1 4 4 6 7