

Задача А. Вперёд!

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

Капрал Дукар любит раздавать приказы своей роте. Самый любимый его приказ — «Вперёд!». Капрал строит солдат в ряд и отдаёт некоторое количество приказов, каждый из которых звучит так: «Рядовые с l_i по l_j — вперёд!»

Перед тем, как Дукар отдал первый приказ, солдаты были пронумерованы от 1 до n слева направо. Услышав приказ «Рядовые с l_i по l_j — вперёд!», солдаты, стоящие на местах с l_i по l_j включительно, продвигаются в начало ряда в том же порядке, в котором были.

Например, если в какой-то момент солдаты стоят в порядке 2, 3, 6, 1, 5, 4, то после приказа «Рядовые с 2 по 4 — вперёд!», порядок будет таким: 3, 6, 1, 2, 5, 4. А если потом Капрал вышлет вперёд солдат с 3 по 4, то порядок будет уже таким: 1, 2, 3, 6, 5, 4.

Вам дана последовательность приказов Капрала. Найдите порядок, в котором будут стоять солдаты после исполнения всех приказов

Формат входных данных

В первой строке входного файла указаны числа n и m ($2 \leq n \leq 10^5$, $1 \leq m \leq 10^5$) число солдат и число приказов. Следующие m строк содержат приказы в виде двух целых чисел: l_i и r_i ($1 \leq l_i \leq r_i \leq n$).

Формат выходных данных

Выведите в выходной файл n целых чисел порядок, в котором будут стоять солдаты после исполнения всех приказов.

Пример

стандартный ввод	стандартный вывод
6 3	1 4 5 2 3 6
2 4	
3 5	
2 2	

Задача В. Переворот

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

Дан массив. Надо научиться обрабатывать два типа запросов.

- 1 $L R$ — перевернуть отрезок $[L, R]$
- 2 $L R$ — найти минимум на отрезке $[L, R]$

Формат входных данных

Первая строка файла содержит два числа n, m . ($1 \leq n, m \leq 10^5$)

Во второй строке находится n чисел a_i ($1 \leq a_i \leq 10^9$) — исходный массив.

Остальные m строк содержат запросы, в формате описанном в условии. Для чисел L, R выполняется ограничение ($1 \leq L \leq R \leq n$).

Формат выходных данных

На каждый запрос типа 2, во входной файл выведите ответ на него, в отдельной строке.

Пример

стандартный ввод	стандартный вывод
10 7	3
5 3 2 3 12 6 7 5 10 12	2
2 4 9	2
1 4 6	2
2 1 8	
1 1 8	
1 8 9	
2 1 7	
2 3 6	

Задача С. Сwoппер

Имя входного файла: `swapper.in`
Имя выходного файла: `swapper.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Перед возвращением в штаб-квартиру корпорации Аазу и Скиву пришлось заполнить на местной таможне декларацию о доходах за время визита. Получилась довольно внушительная последовательность чисел. Обработка этой последовательности заняла весьма долгое время.

— Сwoппер кривой, — со знанием дела сказал таможенник. — А что такое сwoппер? — спросил любопытный Сквив.

Ааз объяснил, что сwoппер — это структура данных, которая умеет делать следующее.

- Взять отрезок чётной длины от x до y и поменять местами число x с $x + 1$, $x + 2$ с $x + 3$, и т.д.
- Посчитать сумму чисел на произвольном отрезке от a до b .

Учитывая, что обсчёт может затянуться надолго, корпорация «МИФ» попросила Вас решить проблему со сwoппером и промоделировать ЭТО эффективно.

Формат входных данных

Во входном файле заданы один или несколько тестов. В первой строке каждого теста записаны число N — длина последовательности и число M — число операций ($1 \leq N, M \leq 10^5$).

Во второй строке теста содержится N целых чисел, не превосходящих 10^6 по модулю — сама последовательность.

Далее следуют M строк — запросы в формате $1\ x_i\ y_i$ — запрос первого типа, и $2\ a_i\ b_i$ — запрос второго типа. Сумма всех N и M по всему файлу не превосходит $2 \cdot 10^5$. Файл завершается строкой из двух нулей. Гарантируется, что $x_i < y_i$, а $a_i \leq b_i$.

Формат выходных данных

Для каждого теста выведите ответы на запросы второго типа, как показано в примере. Разделяйте ответы на тесты пустой строкой.

Пример

<code>swapper.in</code>	<code>swapper.out</code>
5 5	Swapper 1:
1 2 3 4 5	10
1 2 5	9
2 2 4	2
1 1 4	
2 1 3	
2 4 4	
0 0	

Задача D. Задача с региона

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

Во Флатландии протекает богатая рыбой река Большой Флат. Много лет назад река была поделена между n рыболовными предприятиями, каждое из которых получило непрерывный отрезок реки. При этом i -е предприятие, если рассматривать их по порядку, начиная от истока, изначально получило отрезок реки длиной a_i .

С тех пор с рыболовными предприятиями во Флатландии k раз происходили различные события. Каждое из событий было одного из двух типов: банкротство некоторого предприятия или разделение некоторого предприятия на два.

При некоторых событиях отрезок реки, принадлежащий предприятию, с которым это событие происходит, делится на две части. Каждый такой отрезок имеет длину большую или равную 2. Деление происходит по следующему правилу. Если отрезок имеет четную длину, то он делится на две равные части. Иначе он делится на две части, длины которых различаются ровно на единицу, при этом часть, которая ближе к истоку реки, имеет меньшую длину.

При банкротстве предприятия происходит следующее. Отрезок реки, принадлежавший обанкротившемуся предприятию, переходит к его соседям. Если у обанкротившегося предприятия один сосед, то этому соседу целиком передается отрезок реки обанкротившегося предприятия. Если же соседей двое, то отрезок реки делится на две части описанным выше способом, после чего каждый из соседей присоединяет к своему отрезку ближайшую к нему часть.

При разделении предприятия отрезок реки, принадлежавший разделяемому предприятию, всегда делится на две части описанным выше способом. Разделившееся предприятие ликвидируется, и образуются два новых предприятия. Таким образом, после каждого события каждое предприятие владеет некоторым отрезком реки.

Министерство финансов Флатландии предлагает ввести налог на рыболовные предприятия, пропорциональный квадрату длины отрезка реки, принадлежащего соответствующему предприятию. Чтобы проанализировать, как будет работать этот налог, министр хочет по имеющимся данным узнать, как изменялась величина, равная сумме квадратов длин отрезков реки, принадлежащих предприятиям, после каждого произошедшего события.

Требуется написать программу, которая по заданному начальному разделению реки между предприятиями и списку событий, происходивших с предприятиями, определит, чему равна сумма квадратов длин отрезков реки, принадлежащих предприятиям, в начальный момент времени и после каждого события.

Формат входных данных

Первая строка входного файла содержит два целых числа: n и p — исходное количество предприятий ($2 \leq n \leq 10^5$) и номер подзадачи ($0 \leq p \leq 4$).

Вторая строка входного файла содержит n целых чисел a_1, a_2, \dots, a_n — длины исходных отрезков реки.

Третья строка входного файла содержит целое число k — количество событий, происходивших с предприятиями ($1 \leq k \leq 10^5$).

Последующие k строк содержат описания событий, i -я строка содержит два целых числа: e_i и v_i — тип события и номер предприятия, с которым оно произошло. Значение $e_i = 1$ означает, что предприятие, которое после всех предыдущих событий является v_i -м по порядку, если считать с единицы от истока реки, обанкротилось, а значение $e_i = 2$ означает, что это предприятие разделилось на два.

Гарантируется, что значение v_i не превышает текущее количество предприятий. Гарантируется, что если отрезок предприятия при банкротстве или разделении требуется поделить на две части, то он имеет длину большую или равную 2. Гарантируется, что если на реке осталось единственное предприятие, оно не банкротится.

Формат выходных данных

Выходной файл должен содержать $(k + 1)$ целых чисел, по одному в строке. Первая строка должна содержать исходную сумму квадратов длин отрезков реки, а каждая из последующих k строк — сумму квадратов длин отрезков реки после очередного события.

Пример

стандартный ввод	стандартный вывод
4 0	75
3 5 5 4	105
5	73
1 1	101
2 1	83
1 3	113
2 2	
1 3	

Задача Е. Японский кроссворд

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

Маша любит решать японские кроссворды. Так называется головоломка, в которой нужно закрасить некоторые клетки прямоугольника $N \cdot M$ таким образом, чтобы на каждой из N вертикалей и на каждой из M горизонталей количество закрасенных ячеек равнялась некому заранее определенному числу, записанному на полях, для каждой вертикали и горизонтали

К сожалению, иногда составители кроссвордов ошибаются, и кроссворд решения не имеет. Маша не хотела бы тратить время, на решение таких кроссвордов.

Напишите программу, которая для заданных величин N и M , а также $N + M$ чисел, записанных на полях кроссворда, определит, существует ли решение у данного кроссворда.

Формат входных данных

В первой строке входного файла записано число T , ($1 \leq T \leq 6$) — количество кроссвордов для проверки. В следующих $T \cdot 3$ строках идет описание кроссвордов, по три строки на кроссворд.

В первой строке заданы два числа N и M , ($N, M \leq 10^5$) — ширина и высота кроссворда.

Во второй строке записаны N целых, неотрицательных чисел — количество закрасенных клеток в каждом из столбцов с первого по N .

В третьей строке записаны M целых, неотрицательных чисел — количество закрасенных клеток в каждой строчке с первой по M . Ни одно из чисел во второй и третьей строках не превышает общего количества ячеек на соответствующей вертикали или горизонтали.

Формат выходных данных

Выходной файл должен содержать ответ для каждого из кроссвордов в отдельной строке: 1, если кроссворд имеет решение, и 0, если нет. Ответы нужно выводить в том же порядке, в котором во входном файле представлены соответствующие кроссворды.

Пример

стандартный ввод	стандартный вывод
2	1
3 2	0
1 2 1	
2 2	
4 5	
5 5 5 5	
0 0 0 0 0	

Задача F. База данных

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Заведующая учебной частью Байтландской школы в целях контроля успеваемости издала приказ, согласно которому каждый школьник обязан в конце каждой недели предоставлять информацию о своих оценках. Требуется предоставлять лишь свой средний балл, который по мнению завуча полностью характеризует успешность школьника в течение недели.

Родителям понравилось это нововведение, ведь теперь они могут посмотреть уровень успеваемости их ребёнка относительно других школьников. Каждую субботу ученики посылают в базу данных школы свои средние оценки, которые там сохраняются, а затем, родители учеников выполняют m запросов к данной базе данных.

Пусть u — максимальное число, находящееся в базе в данный момент времени, а $cnt(x)$ — количество чисел больше либо равных x (с повторениями), находящихся в базе. Предусмотрены четыре вида запросов к базе данных:

1. Заменить хранимые в базе числа на последовательность $(cnt(1), cnt(2), \dots, cnt(u))$.
2. Добавить в базу некоторое число x .
3. Удалить из базы одно вхождение некоторого числа x , если такое имеется.
4. Вывести количество чисел, равных данному числу x .

Родители начинают запрашивать интересующую их информацию и модернизировать данные лишь после того, как все n школьников отправят свои оценки.

К сожалению, недавно у школы кончилась лицензия на эту замечательную базу данных, поэтому пока её не продлили, на все запросы придётся отвечать вам.

Формат входных данных

Первая строка входных данных содержит два числа n и m ($1 \leq n, m \leq 200\,000$) — количество школьников и количество запросов к базе данных соответственно.

Во второй строке записаны n целых чисел g_i ($1 \leq g_i \leq 200\,000$) — значения средних оценок школьников.

Далее следуют m строк, описывающих запросы в порядке их применения. Каждое описание начинается с одной из букв «t», «a», «r» или «c», означающих запрос первого, второго, третьего или четвертого типа соответственно. Для запросов второго, третьего и четвертого типа далее следует число x_i ($1 \leq x_i \leq 200\,000$) — параметр запроса.

Формат выходных данных

Сначала выведите ответы на все запросы четвертого типа. Затем выведите все числа, находящиеся в базе после выполнения всех запросов, в порядке неубывания.

Гарантируется, что в выходных данных должно будет присутствовать хотя бы одно число.

Пример

стандартный ввод	стандартный вывод
6 8	0
4 3 3 3 6 6	2
t	3 3 5 7 7
c 4	
a 5	
a 3	
r 5	
c 2	
t	
r 3	

Замечание

Рассмотрим как меняется последовательность из примера:

1. (4, 3, 3, 3, 6, 6)
2. (6, 6, 6, 3, 2, 2)
3. (6, 6, 6, 3, 2, 2, 5)
4. (6, 6, 6, 3, 2, 2, 5, 3)
5. (6, 6, 6, 3, 2, 2, 3)
6. (7, 7, 5, 3, 3, 3)
7. (7, 7, 5, 3, 3)

Ниже предоставлены критерии оценки:

№	Баллы	Ограничения	Необх. группы	Комментарий
0	0	—	—	Тесты из условия.
1	15	$n, m, x_i \leq 200$	0	—
2	15	$n, m, x_i \leq 3000$	0 – 1	—
3	35	—	—	Запросы только 1 и 4 типа.
4	35	—	0 – 3	—