

## Задача A. Simple 2D

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

Дан набор точек на плоскости. Напишите структуру данных, отвечающую на запросы «количество точек в прямоугольнике».

### Формат входных данных

В первой строке идёт целое число  $N$  ( $N \leq 10^5$ ) — количество точек.

В последующих  $N$  строках идут координаты точек в формате  $(x_i, y_i)$  — числа, не превосходящие по модулю  $10^9$ .

Далее идёт целое число  $M$  ( $M \leq 10^5$ ) — количество запросов.

В последующих  $M$  строках идут запросы в формате  $(x_{1_j}, y_{1_j}, x_{2_j}, y_{2_j})$  — числа, не превосходящие по модулю  $10^9$ , задающие координаты нижнего-левого и верхнего-правого угла  $j$ -ого прямоугольника-запроса. Стороны считаем принадлежащими многоугольнику.

### Формат выходных данных

Для каждого запроса выведите единственное целое число — количество точек, попавших внутрь прямоугольника.

### Пример

стандартный ввод	стандартный вывод
5	3
-1 -1	5
-1 0	0
0 1	1
0 -1	1
1 -1	
5	
0 -3 3 1	
-1 -1 1 1	
3 4 5 6	
-1 -1 -1 -1	
0 0 0 5	

## Задача В. Вверх [В, А']

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

Когда проблемы тянут ко дну,  
смотри вверх!

Наконец-то дом, который был подвешен к огромному количеству шариков и на котором Карл и Рассел совершили перелет, приземлился на землю. Это путешествие продлилось  $n$  дней. В каждый из дней Рассел записывал насколько изменилась высота дома над землей за этот день.

Лететь над землей безо всякой связи с цивилизацией довольно скучное занятие. Поэтому каждый раз, когда высота дома за день не изменялась, Карл и Рассел очень расстраивались, что день вышел настолько скучным.

Они хотят проанализировать путешествие, поэтому у них есть несколько запросов. Каждый запрос задается некоторым отрезком  $[l, r]$  ( $1 \leq l \leq r \leq n$ ) дней путешествия. Ответом на запрос является максимальное количество подряд идущих скучных дней путешествия начиная с  $l$ -го по  $r$ -й включительно. К сожалению, задача осложняется тем, что иногда Рассел вспоминает, что совершил ошибку и на самом деле в  $i$ -й день высота дома изменилась на  $x$ .

Помогите Карлу и Расселу и напишите программу, отвечающую на их запросы с учетом исправлений ошибок Расселом.

### Формат входных данных

В первой строке написано целое число  $n$  ( $1 \leq n \leq 500\,000$ ) — количество дней, которое продлилось путешествие. Во второй строке записано  $n$  целых чисел  $h_1, h_2, \dots, h_n$  ( $|h_i| \leq 1000$ ) — изначальные изменения высот, записанные Расселом. В третьей строке написано целое число  $m$  ( $1 \leq m \leq 50\,000$ ) — количество запросов. В следующих  $m$  находится описание запросов в следующем формате:

- **QUERY**  $l$   $r$  — вас просят посчитать максимальное количество подряд идущих скучных дней, которое было начиная с  $l$ -го по  $r$ -й дни включительно ( $1 \leq l \leq r \leq n$ );
- **UPDATE**  $i$   $x$  — Рассел нашел ошибку и реальное изменение высоты дома за  $i$ -й день теперь равно  $x$  ( $1 \leq i \leq n, |x| \leq 1000$ ).

### Формат выходных данных

Выведите ответы на все запросы типа **QUERY** в том порядке, в котором они заданы во входном файле.

### Пример

стандартный ввод	стандартный вывод
5	2
328 0 0 0 0	0
6	2
QUERY 1 3	1
UPDATE 3 832	
QUERY 3 3	
QUERY 1 5	
UPDATE 5 -239	
QUERY 1 5	

## Замечание

В первом запросе 2-й и 3-й дни были скучными, поэтому ответ равен 2.

После второго запроса изменения высот в течении путешествия равны 328, 0, 832, 0, 0.

В третьем запросе между 3-м и 3-м днем нету скучных дней, поэтому ответ 0.

В четвертом запросе между 1-м и 5-м днем 2-й, 4-й и 5-й скучные. Среди них есть 2 подряд идущих и это максимальное количество подряд идущих скучных дней, поэтому ответ 2.

## Задача С. Объединение прямоугольников

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

Дано  $N$  прямоугольников со сторонами, параллельными осям координат и вершинами в целочисленных точках. Найдите площадь их объединения.

### Формат входных данных

В первой строке дано число  $0 \leq N < 10^5$  - количество прямоугольников. В следующих  $N$  строках даны описания прямоугольников. Каждое описание прямоугольника — это 4 числа через пробел:  $\langle x_1, y_1, x_2, y_2 \rangle$ . Левый нижний угол прямоугольника имеет координаты  $\langle x_1, y_1 \rangle$ , правый верхний угол имеет координаты  $\langle x_2, y_2 \rangle$ .

$$-10^9 \leq x_1 \leq x_2 \leq 10^9;$$

$$-10^9 \leq y_1 \leq y_2 \leq 10^9$$

### Формат выходных данных

Выведите одно число — площадь объединения этих прямоугольников

### Примеры

стандартный ввод	стандартный вывод
2 0 0 2 2 1 3 2 4	5
0	0
3 1 1 3 5 5 2 7 4 2 4 6 7	23

## Задача D. Бесконечный этаж

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

Знаете ли вы, почему четвертый этаж заперт и там не останавливается лифт? Потому что на самом деле четвертый, запертый, этаж, где не останавливается лифт, содержит бесконечное количество комнат, пронумерованных натуральными числами. На этот этаж регулярно приезжают дети, каждый из которых заранее выбрал, в какую комнату он хочет заселиться. Если выбранная комната оказывается свободна, то ребенок занимает ее, в противном случае он занимает первую свободную комнату с номером, большим исходного.

Кроме того, иногда дети уезжают. Сразу после отъезда ребенка его комната становится доступна для заселения следующего.

Промоделируйте работу вожатых, ответственных за четвертый этаж и научитесь быстро сообщать приезжающим детям, какую комнату им следует занимать.

### Формат входных данных

Первая строка входного файла содержит натуральное число  $n$  — количество прибытий и отъездов ( $1 \leq n \leq 100000$ ).

Следующие  $n$  строк содержат информацию о школьниках. Число  $a > 0$  обозначает, что приехал школьник, желающий занять комнату номер  $a$  ( $1 \leq a \leq 100000$ ). Число  $a < 0$  обозначает, что из комнаты номер  $|a|$  уехал школьник (гарантируется, что эта комната не была пуста).

### Формат выходных данных

Для каждого приезжающего школьника выведите одно натуральное число — номер комнаты, в которую он поселится.

### Пример

стандартный ввод	стандартный вывод
6	5
5	6
5	7
5	6
-6	8
5	
5	

### Замечание

Мы знаем, что вы эту задачу сотней способов решить можете, но напишите спуск в ДО, пожалуйста

## Задача E. Вудист Мирко

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

Юный Мирко решил купить куклу вуду. Учитывая что он крайне заинтересован в том, чтобы купить ее как можно дешевле, он начал отслеживать цены на кукол вуду каждый день. Его список состоит из цен на куклы в последние  $N$  дней, где  $a_i$  обозначает цену куклы  $i$  дней назад.

Мирко думает, что нашел связь между средней ценой кукол в течении нескольких последовательных дней и ценой куклы в следующий день. Он хочет проверить свою догадку и задался вопросом: "Для данного числа  $P$ , сколько существует наборов последовательных дней в течении последних  $N$  дней, для которых средняя цена куклы в эти дни составляет не менее  $P$ ".

Два набора последовательных дней считаются различными, если у них отличается первый или последний день.

### Формат входных данных

Первая строка содержит одно целое число  $N$  ( $1 \leq N \leq 10^6$ ), количество дней в списке Мирко.

Вторая строка содержит  $N$  целых чисел  $a_i$  ( $0 \leq a_i \leq 10^9$ ) — цены кукол в соответствующие дни.

Третья строка содержит одно целое число  $P$  ( $0 \leq P \leq 10^9$ ).

### Формат выходных данных

Выведите одно целое число - ответ на вопрос Мирко для данного  $P$ .

### Примеры

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

## Задача F. Профессиональный декоратор заборов

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

Вася работает подмастерьем в известной студии. Недавно ему поручили помогать молодому, но подающему большие надежды художественному декоратору заборов и изгородей Витезславу Смолокурову. Миссия эта очень ответственная, и от ее выполнения зависит Васино будущее.

Стиль Смолокурова очень необычен, а его работы пользуются большим спросом. Процесс работы разделен на два этапа. На первом этапе Вася делает заготовку — длинный забор, который состоит из набора цветных вертикальных планок. На втором этапе Витезслав приступает к работе.

Для того, чтобы придать забору более спокойный и гармоничный вид, он несколько раз производит следующую операцию: выбирает некоторый цвет и отрезок, после чего перекрашивает этот отрезок забора в выбранный цвет. По своей творческой натуре, Смолокуров может в корне менять концепцию узора по несколько раз за час, поэтому иногда он перекрашивает одну и ту же планку несколько раз. Кроме того, Витезслав не хочет, чтобы какой-то узор повторялся слишком часто. Для того, чтобы избежать этого, он иногда проверяет, не совпадает ли один отрезок забора с другим.

Несложно догадаться, что и перекрашивание, и проверки осуществляет Вася. Работа эта не самая простая, поэтому Вася просит ему помочь хотя бы с проверками на совпадение.

### Формат входных данных

Первая строка входного файла содержит одно целое число  $n$  — количество планок в заборе ( $1 \leq n \leq 100\,000$ ). Вторая строка содержит  $n$  целых чисел, разделенных пробелами — цвета соответствующих планок.

Третья строка входного файла содержит одно целое число  $m$  — количество сравнений и перекрашиваний ( $1 \leq m \leq 100\,000$ ). Следующие  $m$  строк содержат описания заданий, который Вася получает от Витезслава: четыре целых числа  $q, l, r$  и  $k$ .

В случае перекрашивания  $q = 0$ . Эта запись означает перекрашивание всех планок с  $l$  по  $r$  включительно в цвет  $k$  ( $1 \leq l \leq r \leq n$ ). В запросе на сравнение  $q = 1$ . Эта запись означает сравнение кусков забора длины  $k$  начиная с позиций  $l$  и  $r$  соответственно ( $1 \leq l, r \leq n - k + 1, k > 0$ ).

Все числа во входном файле положительные и не превышают 100 000.

### Формат выходных данных

Выведите одну строку: для каждого запроса на сравнение выведите '+' в случае совпадения соответствующих кусков забора и '-' в противном случае.

### Примеры

стандартный ввод	стандартный вывод
7 1 2 1 3 1 2 1 3 0 4 5 2 1 3 1 2 1 3 1 3	+-
2 1 2 5 1 1 2 1 0 2 2 1 1 1 2 1 0 1 2 3 1 1 1 2	--+

## Задача G. Простое задание

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

Это задание очень простое. Вам дана строка  $S$  длины  $n$  и  $q$  запросов, каждый запрос имеет формат  $i j k$ , что означает: отсортировать подстроку, состоящую из символов от  $i$  до  $j$ , в неубывающем порядке, если  $k = 1$  или в невозрастающем порядке, если  $k = 0$ .

Выведите итоговую строку после выполнения запросов.

### Формат входных данных

В первой строке записано два целых числа  $n, q$  ( $1 \leq n \leq 10^5, 0 \leq q \leq 50000$ ), длина строки и количество запросов, соответственно.

В следующей строке идёт сама строка  $S$ . Она состоит только из строчных английских букв.

В каждой из следующих  $q$  строк записано по три целых числа  $i, j, k$  ( $1 \leq i \leq j \leq n, 0 \leq k \leq 1$ ), обозначающих запрос.

### Формат выходных данных

Выведите строку  $S$  после выполнения всех запросов.

### Пример

стандартный ввод	стандартный вывод
10 10 ittmcsvmoa 6 7 0 2 4 0 4 10 1 1 2 0 2 9 1 5 6 1 7 9 0 1 2 0 2 6 0 3 9 1	tmacimostv

## Задача Н. Эффективное тестирование

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

Начиная с 20xx года все организаторы всех школьных олимпиад по программированию договорились проводить соревнования исключительно по интернету, для чего было создано общество с ограниченной ответственностью «Организация онлайн-олимпиад» (ООО «ООО»). Разумеется, такая серьёзная организация не может обойтись без собственной тестирующей системы, поэтому для её создания были наняты эффективные менеджеры, закуплены доски и подготовлена синяя изолента.

Для повышения эффективности процесса тестирования была разработана следующая архитектура. Сначала все  $m$  тестов задачи располагаются в порядке от 1 к  $m$  в очереди тестирования. Затем модуль планирования последовательно выполняет  $n$  действий. Действие  $i$  состоит в том, чтобы выбрать отрезок очереди с позиции  $l_i$  по  $r_i$  включительно (в нумерации с единицы) и проверить решение на каждом втором тесте на этом отрезке, а именно на тестах на позициях  $l_i, l_i+2, l_i+4, \dots, r_i$  очереди (при этом гарантируется, что  $l_i$  и  $r_i$  имеют одинаковую чётность). После этого те тесты, на которых было проведено тестирование, удаляются из очереди, а все оставшиеся тесты сдвигаются по очереди таким образом, чтобы пустых мест не осталось. Например, если в очереди находились тесты с исходными номерами 2, 3, 4, 5, 10, 12, 13, 20 и была применена операция с  $l_i = 3, r_i = 7$ , то посылка будет протестирована на тестах с позиций 3, 5 и 7, которые исходно имели номера 4, 10 и 13. После выполнения данной операции очередь тестирования будет состоять из тестов с исходными номерами 2, 3, 5, 12, 20.

Вам поручено реализовать модуль, который для каждого из  $n$  описанных выше действий будет определять минимальный и максимальный номер теста в изначальной нумерации из тех, на которых на этом шаге проверялось решение.

### Формат входных данных

В первой строке входных данных находятся два числа  $n$  и  $m$  ( $1 \leq n \leq 100\,000, 1 \leq m \leq 10^{18}$ ) — количество действий модуля планирования и количество тестов в задаче.

В каждой из последующих  $n$  строк записаны два целых числа  $l_i$  и  $r_i$  ( $1 \leq l_i \leq r_i \leq m$ ) — параметры  $i$ -го действия модуля планирования. Гарантируется, что перед началом выполнения действия  $i$  в очереди тестирования находятся хотя бы  $r_i$  тестов и что числа  $l_i$  и  $r_i$  имеют одинаковую чётность.

### Формат выходных данных

Для каждого из  $n$  действий модуля планирования выведите два целых числа — минимальный и максимальный номер теста в исходной нумерации из тех, на которых проверялось решение на соответствующем шаге.

### Примеры

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

### Замечание

Рассмотрим, как изменяется очередь тестирования в первом примере.

1. Изначально в очереди тестирования находятся все тесты от 1 до 10, то есть очередь имеет вид 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.

2. При выполнении первого запроса будут удалены тесты 2, 4, 6, 8, и очередь примет вид 1, 3, 5, 7, 9, 10.
3. При выполнении второго запроса будут удалены тесты 1 и 5, очередь примет вид 3, 7, 9, 10.

## Задача I. Фэйк Ньюз

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

В Байтландии зарегистрировано несколько информационных агентств, в том числе новое информационное агентство ООО «неверлай дот ком». Начальник этой организации хочет выпустить сенсационное расследование. Последний месяц он ждет подходящего момента, чтобы новость стала как можно более популярной.

У каждого агентства есть свой показатель авторитетности  $A$ , который может отличаться для разных агентств. Процесс публикации новости в Байтландии устроен так: первым делом агентство ООО «неверлай дот ком» публикует новость. Это запускает цепную реакцию, устроенную следующим образом:

- Каждое агентство из тех, кто еще не опубликовал данную новость, вычисляет число  $P$ : количество других агентств, опубликовавших новость.
- Если агентство имеет показатель авторитетности  $A$ , а число уже опубликовавших новость агентств  $P$  не меньше  $A$ , то это агентство публикует данную новость.
- Процесс повторяется до тех пор, пока предыдущее условие выполняется хотя бы для одного из агентств.

ООО «неверлай дот ком» хочет выбрать момент, когда им стоит опубликовать новость, но в Байтландии часто появляются новые информационные агентства, а старые часто закрываются. От вас требуется обрабатывать запросы двух типов:

1. «+  $A$ » появляется новое агентство с показателем авторитетности  $A$
2. «-  $A$ » одно из агентств с показателем авторитетности  $A$  прекращает работу.

Для запросов второго типа гарантируется, что существовало хотя бы одно СМИ с показателем авторитетности  $A$ . Обратите внимание, что в один момент может существовать несколько агентств с одинаковым показателем авторитетности  $A$ .

После каждого запроса вам требуется определить, сколько агентств опубликуют новость, если ООО «неверлай дот ком» опубликует ее прямо сейчас.

### Формат входных данных

В первой строке задано целое число  $n$  ( $1 \leq n \leq 500\,000$ ) — количество запросов.

В следующих  $n$  строках вводятся запросы в формате, описанном в условии задачи: сначала задаётся один из символов «+», «-», затем следует целое число  $A$  — показатель авторитетности ( $1 \leq A \leq 500\,000$ ).

### Формат выходных данных

Выведите  $n$  строк. В каждой строке должно быть одно число — количество агентств, которые опубликуют новость от ООО «неверлай дот ком» после  $i$ -го запроса.

## Примеры

стандартный ввод	стандартный вывод
5	1
+ 4	2
+ 1	2
+ 3	5
+ 2	3
- 3	
8	2
+ 1	3
+ 1	4
+ 2	3
- 1	3
+ 4	1
- 1	3
+ 1	5
+ 1	

## Замечание

Рассмотрим первый пример:

- После первого запроса ни одно другое агенство не будет публиковать новость от ООО «неверлай дот ком».
- После второго запроса появилось агенство, которое готово опубликовать новость.
- После третьего запроса все еще лишь одно агенство готово опубликовать новость.
- После четвёртого запроса процесс публикации будет устроен следующим образом:
  1. ООО «неверлай дот ком» публикует новость.
  2. агенство с  $A = 1$  публикует новость.
  3. агенство с  $A = 2$  публикует новость.
  4. агенство с  $A = 3$  публикует новость.
  5. агенство с  $A = 4$  публикует новость.
- После последнего запроса процесс публикации остановится на агенстве с  $A = 2$ .