Задача A. Множества

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	0.75 секунд
Ограничение по памяти:	256 мегабайт

На вступительном контесте в пилотную группу по программированию Вашему другу предложили реализовать структуру данных для хранения множеств чисел. Так как он специализируется на истории литературы, данную структуру придётся реализовать Вам.

Структура должна хранить $m+1$ множеств чисел от 0 до n, пронумерованных от 0 до m включительно, при этом одно число может принадлежать сразу нескольким множествам. Изначально все множества пустые.

Вы должны реализовать следующие операции на этой структуре:

1. `ADD e s`

   Добавить в множество №s ($0 \leqslant s \leqslant m$) число e ($0 \leqslant e \leqslant n$).

2. `DELETE e s`

   Удалить из множества №s ($0 \leqslant s \leqslant m$) число e ($0 \leqslant e \leqslant n$). Гарантируется, что до этого число e было помещено в множество

3. `CLEAR s`

   Очистить множество №s ($0 \leqslant s \leqslant m$).

4. `LISTSET s`

   Показать содержимое множества №s ($0 \leqslant s \leqslant m$) **в возрастающем порядке**, либо -1, если множество пусто.

5. `LISTSETSOF e`

   Вывести номера множеств, в которых лежит число e ($0 \leqslant e \leqslant n$) **в возрастающем порядке**, либо -1, если этого числа нет ни в одном множестве.

Формат входных данных

Сначала вводятся числа N ($1 \leqslant N \leqslant 9223372036854775807$), M ($1 \leqslant M \leqslant 100000$) и K ($0 \leqslant K \leqslant 100000$) — максимальное число, номер максимального множества и количество запросов к структуре данных. Далее следуют K строк указанного формата запросов.

Формат выходных данных

На каждый запрос `LISTSET` Ваша программа должна вывести числа — содержимое запрошенного множества или -1, если множество пусто.

На каждый запрос `LISTSETSOF` Ваша программа должна вывести числа — номера множеств, содержащих запрошенное число, или -1, если таких множеств не существует.

На прочие запросы не должно быть никакого вывода.

Гарантируется, что правильный вывод программы не превышает одного мегабайта.

Пример

стандартный ввод	стандартный вывод
10 10 9 ADD 1 1 ADD 1 2 ADD 2 1 LISTSET 1 LISTSETSOF 1 DELETE 1 1 LISTSET 1 CLEAR 1 LISTSET 1	1 2 1 2 2 -1

Замечание

Эту задачу можно (и нужно!) решать, используя std::set и std::map.

Задача B. Гоблины и очереди

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	0.6 секунд
Ограничение по памяти:	256 мегабайт

Гоблины Мглистых гор очень любях ходить к своим шаманам. Так как гоблинов много, к шаманам часто образуются очень длинные очереди. А поскольку много гоблинов в одном месте быстро образуют шумную толку, которая мешает шаманам проводить сложные медицинские манипуляции, последние решили установить некоторые правила касательно порядка в очереди.

Обычные гоблины при посещении шаманов должны вставать в конец очереди. Привилегированные же гоблины, знающие особый пароль, встают ровно в ее середину, причем при нечетной длине очереди они встают сразу за центром.

Так как гоблины также широко известны своим непочтительным отношением ко всяческим правилам и законам, шаманы попросили вас написать программу, которая бы отслеживала порядок гоблинов в очереди.

Формат входных данных

В первой строке входных данный записано число N ($1 \leqslant N \leqslant 10^5$) — количество запросов. Следующие N строк содержат описание запросов в формате:

- $+\ i$ — гоблин с номером i ($1 \leqslant i \leqslant N$) встаёт в конец очереди.
- $*\ i$ — привилегированный гоблин с номером i встает в середину очереди.
- `-` — первый гоблин из очереди уходит к шаманам. Гарантируется, что на момент такого запроса очередь не пуста.

Формат выходных данных

Для каждого запроса типа - программа должна вывести номер гоблина, который должен зайти к шаманам.

Примеры

стандартный ввод	стандартный вывод
7 + 1 + 2 - + 3 + 4 - -	1 2 3
2 * 1 + 2	

Задача C. Машинки

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

Петя, которому три года, очень любит играть с машинками. Всего у Пети N различных машинок, которые хранятся на полке шкафа так высоко, что он сам не может до них дотянуться. Одновременно на полу комнаты может находиться не более K машинок. Петя играет с одной из машинок на полу и если он хочет поиграть с другой машинкой, которая также находится на полу, то дотягивается до нее сам. Если же машинка находится на полке, то он обращается за помощью к маме. Мама может достать для Пети машинку с полки и одновременно с этим поставить на полку любую машинку с пола. Мама очень хорошо знает своего ребенка и может предугадать последовательность, в которой Петя захочет играть с машинками. При этом, чтобы не мешать Петиной игре, она хочет совершить как можно меньше операций по подъему машинки с пола, каждый раз правильно выбирая машинку, которую следует убрать на полку. Ваша задача состоит в том, чтобы определить минимальное количество операций. Перед тем, как Петя начал играть, все машинки стоят на полке.

Формат входных данных

В первой строке содержаться три числа N, K и P ($1 \leqslant K, N \leqslant 100000$, $1 \leqslant P \leqslant 500000$). В следующих P строках записаны номера машинок в том порядке, в котором Петя захочет играть с ними.

Формат выходных данных

Выведите единственное число: минимальное количество операций, которое надо совершить Петиной маме.

Пример

стандартный ввод	стандартный вывод
3 2 7 1 2 3 1 3 1 2	4

Замечание

В этой задаче можно использовать STL.

Пояснения к примеру:

Операция 1: снять машинку 1

Операция 2: снять машинку 2

Операция 3: поднять машинку 2 и снять машинку 3

Операция 4: поднять машинку 3 или 1 и снять машинку 2

Задача D. Сортировка вагонов

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

К тупику со стороны пути 1 (см. рисунок) подъехал поезд. Разрешается отцепить от поезда один или сразу несколько первых вагонов и завезти их в тупик (при желании, можно даже завезти в тупик сразу весь поезд). После этого часть из этих вагонов вывезти в сторону пути 2. После этого можно завезти в тупик еще несколько вагонов и снова часть оказавшихся вагонов вывезти в сторону пути 2. И так далее (так, что каждый вагон может лишь один раз заехать с пути 1 в тупик, а затем один раз выехать из тупика на путь 2). Заезжать в тупик с пути 2 или выезжать из тупика на путь 1 запрещается. Нельзя с пути 1 попасть на путь 2, не заезжая в тупик.

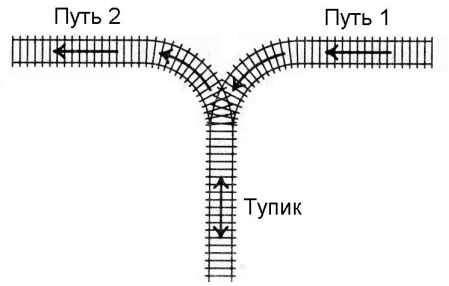


Рис. 1: схема путей

Известно, в каком порядке изначально идут вагоны поезда. Требуется с помощью указанных операций сделать так, чтобы вагоны поезда шли по порядку (сначала первый, потом второй и т.д., считая от головы поезда, едущего по пути 2 в сторону от тупика).

Формат входных данных

Вводится число N — количество вагонов в поезде ($1 \leqslant N \leqslant 2000$). Дальше идут номера вагонов в порядке от головы поезда, едущего по пути 1 в сторону тупика. Вагоны пронумерованы натуральными числами от 1 до N, каждое из которых встречается ровно один раз.

Формат выходных данных

Если сделать так, чтобы вагоны шли в порядке от 1 до N, считая от головы поезда, когда поезд поедет по пути 2 из тупика, можно, выведите действия, которые нужно проделать с поездом. В первой строке выведите количество действий, а затем сами действия. Каждое из них описывается двумя числами: типом и количеством вагонов:

- если нужно завезти с пути 1 в тупик K вагонов, должно быть выведено сначала число 1, а затем — число K ($K \geqslant 1$),
- если нужно вывезти из тупика на путь 2 K вагонов, должно быть выведено сначала число 2, а затем — число K ($K \geqslant 1$).

Если возможно несколько последовательностей действий, приводящих к нужному результату, выведите любую из них.

Если выстроить вагоны по порядку невозможно, выведите одно число 0.

Примеры

стандартный ввод	стандартный вывод
3 3 2 1	2 1 3 2 3
4 4 1 3 2	4 1 2 2 1 1 2 2 3

Задача E. Вычислительная ихтиология

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

Игорь работает младшим лаборантом в НИИ ихтиологии. Ему вверены n аквариумов, стоящих в ряд, в каждом из которых живет колония рыбок гуппи. Про каждую колонию заранее известна ее численность.

В лабораторных условиях НИИ ихтиологии колония рыбок гуппи растет по следующему правилу: достигнув популяции в f рыбок, колония живет в течение $max(1000 - f, 1)$ секунд, после чего на свет появляется новая рыбка. От начального момента времени до рождения первой рыбки колония размера f также ждет $max(1000 - f, 1)$ секунд.

Например, колония с начальным размером 996 будет размножаться следующим образом:

время	число рыб	время до очередной рыбки
0	996	4
4	997	3
7	998	2
9	999	1
10	1000	1
11	1001	1
...	...	...

Появление на свет каждой новой рыбки Игорь должен фиксировать в специальном журнале. Будем считать, что запись он делает мгновенно, но при этом он должен в момент рождения новой рыбки находиться рядом с аквариумом, в котором это произошло.

На перемещение от одного аквариума к соседнему у Игоря уходит одна секунда. В начальный момент времени Игорь стоит около первого аквариума.

Вычислите, в течение какого наибольшего периода времени Игорь сможет добросовестно выполнять свою работу.

Формат входных данных

В первой строке входного файла содержится целое число n ($2 \leqslant n \leqslant 50$) — количество аквариумов с рыбками гуппи в НИИ ихтиологии. Каждая из следующих n строк содержит одно целое число a_i ($1 \leqslant a_i 2007$) — численность i-й колонии.

Формат выходных данных

В выходной файл выведите момент времени, когда родится первая рыбка гуппи, запись о рождении которой Игорь сделать не сможет.

Пример

стандартный ввод	стандартный вывод
3 996 1 994	7

Замечание

В приведенном примере Игорь сначала ждет у первого аквариума появления рыбки на 4-й секунде. После этого он бежит к третьему аквариуму (на это у него уходит 2 секунды) и как раз успевает к рождению рыбки на 6-й секунде. Однако вернуться к первому аквариуму, где следующая рыбка родится на 7-й секунде, он уже не успевает.

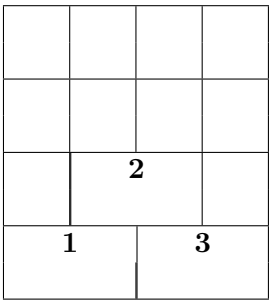
Задача F. Тетрис

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

Как и в обычном тетрисе, поле в игре Strategy Tetris представляет собой «стакан» шириной в W клеток ($1 \leqslant W \leqslant 10^9$) и бесконечной высоты. В этот стакан падают сверху N фигурок ($1 \leqslant N \leqslant 100000$). i-я фигурка представляет собой прямоугольник шириной в W_i клеток и высотой в одну клетку; самая левая клетка фигурки имеет абсциссу a_i ($1 \leqslant a_i \leqslant W - -W_i + 1$). Фигурки падают по обычным правилам: если при падении фигурка хотя бы одной своей клеткой ложится на какую-либо уже упавшую фигурку, то ее движение прекращается.

В отличие от обычного тетриса, игрок не имеет возможности вращать фигурки или смещать их по горизонтали в процессе падения — еще бы, это пришлось бы делать быстро и не было бы времени серьёзно подумать над стратегией. Единственное, что он может — это выбрать порядок, в котором эти N фигурок упадут в стакан (каждая по одному разу). Ваша задача — помочь ему выбрать такой порядок, при котором высота образовавшейся в результате падения конструкции была бы как можно меньше. (В отличие от обычного тетриса, полностью заполненная фигурками горизонталь никуда не исчезает).

На рисунке ниже приведен пример заполнения стакана фигурками из примера входных данных (порядок заполнения соответствует выходному файлу, приведенному в примере).



Формат входных данных

В первой строке входного файла записаны числа N и W, а в последующих N строках — пары чисел a_i и W_i.

Формат выходных данных

Выведите в выходной файл минимальную возможную высоту конструкции, а затем последовательность номеров фигурок, к этой высоте приводящую. Фигурки нумеруются натуральными числами от 1 до N в том порядке, в котором они указаны во входных данных. Если возможных вариантов несколько, выведите любой из них.

Пример

стандартный ввод	стандартный вывод
3 4 1 2 2 2 3 2	2 1 3 2

Задача G. Менеджер памяти

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

Пете поручили написать менеджер памяти для новой стандартной библиотеки языка H++. В распоряжении у менеджера находится массив из N последовательных ячеек памяти, пронумерованных от 1 до N. Задача менеджера — обрабатывать запросы приложений на выделение и освобождение памяти.

Запрос на выделение памяти имеет один параметр K. Такой запрос означает, что приложение просит выделить ему K последовательных ячеек памяти. Если в распоряжении менеджера есть хотя бы один свободный блок из K последовательных ячеек, то он обязан в ответ на запрос выделить такой блок. При этом непосредственно перед самой первой ячейкой памяти выделяемого блока не должно располагаться свободной ячейки памяти. После этого выделенные ячейки становятся занятыми и не могут быть использованы для выделения памяти, пока не будут освобождены. Если блока из K последовательных свободных ячеек нет, то запрос отклоняется.

Запрос на освобождение памяти имеет один параметр T. Такой запрос означает, что менеджер должен освободить память, выделенную ранее при обработке запроса с порядковым номером T. Запросы нумеруются, начиная с единицы. Гарантируется, что запрос с номером T — запрос на выделение, причем к нему еще не применялось освобождение памяти. Освобожденные ячейки могут снова быть использованы для выделения памяти. Если запрос с номером T был отклонен, то текущий запрос на освобождение памяти игнорируется.

Требуется написать менеджер памяти, удовлетворяющий приведенным критериям.

Формат входных данных

В первой строке входных данных задаются числа N и M — количество ячеек памяти и количество запросов, соответственно ($1 \leqslant N \leqslant 2^{31} - -1$; $1 \leqslant M \leqslant 10^5$). Каждая из следующих M строк содержит по одному числу: $(i+1)$-я строка входных данных ($1 \leqslant i \leqslant M$) содержит либо положительное число K, если i-й запрос — запрос на выделение с параметром K ($1 \leqslant K \leqslant N$), либо отрицательное число $-T$, если i-й запрос — запрос на освобождение с параметром T ($1 \leqslant T < i$).

Формат выходных данных

Для каждого запроса на выделение памяти выведите результат обработки этого запроса: для успешных запросов выведите номер первой ячейки памяти в выделенном блоке, для отклоненных запросов выведите число -1. Результаты нужно выводить в порядке следования запросов во входных данных

Пример

стандартный ввод	стандартный вывод
42 9 7 3 8 -2 6 5 -5 9 4	1 8 11 19 25 30 19