

Задача А. Посевные работы

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

На дворе уже ранняя весна, а Бомбослав только что получил диплом заключительного этапа Открытой олимпиады школьников по программированию и сейчас находится на даче, помогая своему дедушке Пафнутию Максимовичу засеять огород.

Огород представляет собой прямоугольник размеров $w \times h$, где w и h — целые числа. В этом году Пафнутий Максимович решил засеять огород репой и редькой. При этом он хочет для каждого из растений выделить некоторую область огорода в виде квадрата с целой длиной стороны, причём ориентированного таким образом, что стороны квадрата параллельны сторонам огорода. Квадраты могут иметь разные размеры, но они не должны выходить за пределы огорода и не могут пересекаться по ненулевой площади.

Бомбослав хочет продемонстрировать дедушке, что не зря получил диплом Олимпиады, и поэтому он решил вычислить, какую максимальную площадь огорода можно засеять указанным способом. Поскольку репу дедушка Бомбослава любит меньше, чем редьку, разрешается не сеять её вовсе (то есть использовать только одну квадратную область), если это позволит покрыть бóльшую площадь огорода.

Формат входных данных

В единственной строке входных данных записаны два целых числа w и h ($1 \leq w, h \leq 10^9$) — размеры огорода.

Формат выходных данных

Выведите одно целое число — максимальную площадь огорода, которую можно засеять, выделив одну или две квадратные области описанного в условии вида с максимально возможной суммарной площадью.

Примеры

стандартный ввод	стандартный вывод
2 2	4
4 3	10

Замечание

В первом примере Бомбослав может использовать одну квадратную область 2×2 , которой он покроет весь огород. Во втором примере Бомбослав использует одну квадратную область 3×3 , которой он покроет квадрат 1×1 .

Система оценки

В данной задаче три группы тестов. Тесты оцениваются **независимо**, прохождение каждого теста (кроме теста из условия) оценивается в 2 балла. Результаты проверки всех тестов доступны во время соревнования.

Группа	Баллы	Дополнительные ограничения	Комментарий
		w, h	
0	0	—	Тесты из условия.
1	30	$w, h \leq 5$	
2	20	$w, h \leq 100$	
3	20	$w, h \leq 10\,000$	
4	30	—	

Задача В. В поисках максимальной суммы

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

Мама подарила маленькой девочке Алёне массив чисел. Девочку заинтересовали непрерывные подмассивы с равными числами на концах. Среди таких подмассивов ненулевой длины Алёна хочет найти подмассив с максимальной суммой. Формально говоря, требуется найти такие $1 \leq l \leq r \leq n$, что $a_l = a_r$ и сумма чисел $a_l + a_{l+1} + \dots + a_r$ максимальна.

Формат входных данных

В первой строке входных данных находится число n ($1 \leq n \leq 1\,000\,000$) — количество чисел в массиве a .

Во второй строке входных данных находятся n целых чисел a_1, a_2, \dots, a_n ($-10^9 \leq a_i \leq 10^9$).

Формат выходных данных

В первой строке выведите максимальную сумму в подмассиве, удовлетворяющем условию задачи.

Во второй строке выведите 2 целых числа l и r , такие что $1 \leq l \leq r \leq n$ и a_l, a_{l+1}, \dots, a_r — искомый подмассив с максимальной суммой.

Если существует несколько ответов, выведите любой из них.

Примеры

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

Замечание

Обратите внимание, во втором примере все числа отрицательные, но Алёна всё равно должна выбрать какой-то непустой подмассив.

Система оценки

Тесты к этой задаче состоят из четырёх групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **предыдущих** групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Баллы	Дополнительные ограничения		Комментарий
		n	a_i	
0	0	—	—	Тесты из условия.
1	30	$n \leq 500$	$ a_i \leq 1\,000\,000$	
2	30	$n \leq 5000$	$ a_i \leq 1\,000\,000$	
3	40	—	—	Offline-проверка.

Задача С. Жезл Всевластия

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

Колдун Азазель заполучил наконец Жезл Всевластия, на который последовательно нанизаны волшебные кольца. На каждом кольце выгравирована одна строчная или заглавная буква английского алфавита. Таким образом, если прочитать буквы на кольцах от начала к концу жезла, то получится строка s . Перед использованием своего нового сокровища на батле всех колдунов Азазель решил изучить его магические свойства.

Прочитав прилагавшуюся к Жезлу Всевластия инструкцию, Азазель узнал, что магические свойства жезла раскрываются только если в строке s не встречается в качестве **подпоследовательности** заклинание, определяемое строкой t . Строка t встречается в строке s в качестве подпоследовательности, если из строки s можно получить строку t , выкинув некоторые буквы (возможно, пустое множество). Обратите внимание, что оставшиеся буквы не обязаны следовать подряд в исходной строке s .

Теперь Азазель хочет снять с жезла некоторые кольца, так чтобы строка t не встречалась в нём в качестве подпоследовательности. Поскольку он не додумался прочитать инструкцию до конца, то оставшиеся кольца он разместит на жезле в **том же самом порядке**, в котором они шли исходно. Чтобы сохранить у жезла как можно больше магической силы, Азазель планирует снять как можно меньшее количество колец.

Формат входных данных

В первой строке входных данных записана строка s ($1 \leq |s| \leq 10\,000$, где $|s|$ обозначает длину строки s), определяющая буквы на соответствующих кольцах в порядке от начала к концу жезла. Во второй строке записана строка t ($2 \leq |t| \leq 1000$, $|t| \leq |s|$) — заклинание, которое не должно встречаться на жезле в качестве подпоследовательности.

Гарантируется, что обе строки состоят только из строчных и заглавных букв английского алфавита.

Формат выходных данных

Выведите строку, которая получится при чтении от начала к концу жезла букв, написанных на кольцах, которые должен оставить Азазель. Если подходящих ответов максимальной длины несколько, разрешается вывести любой из них.

Гарантируется, что Азазель всегда сможет оставить на жезле хотя бы одно кольцо.

Примеры

стандартный ввод	стандартный вывод
abacaba aba	aacab
AbCa Aba	AbC

Замечание

В первом примере строка «bacaа» также является правильным ответом.

Система оценки

Тесты к этой задаче состоят из восьми групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых предыдущих групп, указанных в таблице ниже. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Баллы	Доп. ограничения		Необх. группы	Комментарий
		$ s $	$ t $		
0	0	–	–	–	Тесты из условия.
1	30	$ s \leq 15$	$ t \leq 15$	0	
2	20	$ s \leq 100$	$ t \leq 52$	–	Все буквы строки t различны.
3	20	$ s \leq 500$	$ t \leq 500$	0, 1, 2	
4	30	–	–	0, 1, 2, 3	Offline-проверка.

Задача D. Занимательный эксперимент

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Маленький Дима увлекается физикой и обожает эксперименты. Сегодня он решил поставить очередной познавательный опыт.

У Димы есть бесконечно высокая труба, на дне которой находится клапан, позволяющий при активации уменьшить уровень воды в трубе ровно на один сантиметр. Внутри трубы расположены n датчиков. Датчик с номером i находится на высоте h_i сантиметров от дна трубы, при этом датчик с бóльшим номером находится на большей высоте. Все датчики подключены к цепи управления клапаном, которая опрашивает датчики по очереди, начиная с датчика с номером n и заканчивая датчиком с номером 1, при этом датчик с номером i опрашивается v_i раз подряд. Во время опроса датчика с номером i , если уровень воды не ниже высоты, на которой находится датчик, открывается клапан, и уровень воды понижается на один сантиметр.

Дима решил проводить эксперимент следующим образом.

1. Дима выбирает некоторое целое число x .
2. В начале каждой секунды Дима заливает в трубу воду так, чтобы уровень воды в трубе повысился на x сантиметров.
3. Затем в эту же секунду Дима запускает цепь управления, в результате чего уровень воды понижается.
4. Если в конце секунды уровень воды не меньше, чем H сантиметров, то Дима заканчивает эксперимент. В противном случае Дима продолжает эксперимент, заливая в трубу ещё воды в начале следующей секунды.

Считайте, что процесс добавления воды и работы цепи управления занимает пренебрежимо малое время. Также считайте, что труба достаточно высокая, чтобы вместить любое количество воды. В начале эксперимента воды в трубе нет.

Через T секунд Димина мама вернётся домой и будет очень недовольна, если увидит, как Дима занимается переливанием воды вместо решения задач заочного тура Открытой олимпиады по программированию. Поэтому Дима решил выбрать x так, чтобы эксперимент закончился не позже, чем через T секунд. Кроме того, Дима не хочет носить много воды и хочет выбрать минимальное подходящее x . Помогите Диме успеть провести эксперимент до возвращения мамы домой.

Формат входных данных

В первой строке входных данных заданы три целых числа n , H и T ($1 \leq n \leq 100\,000$, $1 \leq H, T \leq 10^9$) — количество датчиков, требуемый уровень воды и время до прихода мамы соответственно.

Следующие n строк описывают имеющиеся в трубе датчики. В i -й из этих строк находятся два целых числа h_i и v_i ($1 \leq h_i \leq H$, $1 \leq v_i \leq 10^9$) — высота, на которой находится датчик с номером i и количество раз, которое цепь управления опрашивает этот датчик.

Гарантируется, что $1 \leq h_1 < h_2 < \dots < h_n \leq H$.

Формат выходных данных

Выведите одно целое число — минимальное значение величины x , которое позволит Диме завершить эксперимент за не более чем T секунд.

Примеры

стандартный ввод	стандартный вывод
1 10 5 8 4	5
1 50 4 30 60	67
2 7 2 1 2 7 3	8
3 15 3 1 5 6 1 15 1	12

Замечание

В первом тестовом примере эксперимент происходит следующим образом.

1. В начале первой секунды Дима зальёт воду и уровень станет равен 5 сантиметрам. Так как вода находится ниже единственного датчика, во время работы цепи он ни разу не сработает и уровень воды не изменится.
2. В начале второй секунды уровень воды станет равен 10 сантиметрам. После этого датчик будет опрошен 4 раза. Перед первым, вторым и третьим опрашиванием уровень воды в трубе будет составлять 10, 9 и 8 сантиметров соответственно, поэтому вода будет сливаться каждый раз. Перед четвёртым опрашиванием уровень воды будет находиться на отметке в 7 сантиметров и клапан активирован не будет, а значит, уровень воды не изменится.
3. В начале третьей секунды Дима поднимет уровень воды до 12 сантиметров, после чего датчик сработает четыре раза подряд и уровень воды упадёт до 8 сантиметров.
4. В начале четвёртой секунды Дима поднимает уровень воды до 13 сантиметров, датчик сработает четыре раза и уровень воды опустится до 9 сантиметров.
5. В начале пятой секунды Дима поднимет уровень воды до 14 сантиметров, датчик сработает четыре раза, уровень воды понизится до 10 сантиметров. Таким образом Дима закончит эксперимент за 5 секунд, как раз в момент когда мама будет на пороге квартиры.

Можно проверить, что ни при каком меньшем значении x Дима не успеет закончить опыт до прихода мамы.

Рассмотрим ход эксперимента в четвёртом тестовом примере.

1. В начале первой секунды Дима зальёт воду в трубу, и уровень воды установится на отметке в 12 сантиметров. Датчик на высоте 15 не сработает, датчик на высоте 6 сработает один раз, в результате чего уровень воды понизится до 11 сантиметров. После этого датчик на высоте 1 сработает 5 раз и уровень воды станет равным 6.
2. В начале второй секунды Дима поднимет уровень воды до 18 сантиметров. После этого датчики на высотах 15 и 6 сработают по разу, а датчик на высоте 1 — 5 раз, в результате чего клапан будет активирован 7 раз, и уровень воды снизится до 11 сантиметров.
3. В начале третьей секунды Дима зальёт воду до отметки в 23 сантиметра. Как и в прошлую секунду, датчики сработают 7 раз, понижая уровень воды до 16 сантиметров, после чего Дима завершит свой эксперимент.

Система оценки

Тесты к этой задаче состоят из шести групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **предыдущих** групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Баллы	Дополнительные ограничения			Комментарий
		n	H, h_i, v_i	T	
0	0	–	–	–	Тесты из условия.
1	10	$n = 1$	$H, h_i, v_i \leq 100$	$T \leq 100$	
2	15	$n = 2$	$H, h_i, v_i \leq 100$	$T \leq 100$	
3	15	$n \leq 100$	$H, h_i, v_i \leq 100$	$T \leq 100$	
4	15	$n \leq 3000$	$H, h_i, v_i \leq 3000$	$T \leq 3000$	
5	15	$n \leq 3000$	–	$T \leq 3000$	
6	30	–	–	–	Offline-проверка.

Задача Е. Задержки рейсов

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

Алёна отправляется в долгожданный заслуженный отпуск. Алёна очень боится опоздать на самолёт, поэтому она приехала в аэропорт в момент времени 0. На табло отправления отображается n рейсов, рейс номер i (в нумерации с единицы) должен вылететь в момент времени t_i , при этом $t_i > t_{i-1}$ для всех $i \geq 2$.

К сожалению, погода в городе N очень непредсказуемая, и именно сейчас в районе аэропорта начался сильный снегопад. Для обеспечения безопасности, службы аэропорта должны после отправления каждого рейса снова очищать взлётно-посадочную полосу от снега, поэтому времена отправления двух последовательных рейсов должны отличаться хотя бы на k .

Алёна знает, что рейсы будут отправляться в порядке возрастания своих номеров, при этом каждый рейс будет вылетать настолько рано, насколько это возможно, но не менее чем через k после вылета предыдущего рейса и не ранее изначально запланированного времени отправления t_i .

Поскольку мысли Алёны уже целиком посвящены отпуску, то написать программу, которая определит реальное время отправления каждого из рейсов, предстоит вам. Считайте, что для рейса с номером 1 нет предыдущего и для него не требуется чистить взлётно-посадочную полосу.

Формат входных данных

В первой строке входных данных записаны два целых числа n и k ($1 \leq n \leq 100\,000$, $1 \leq k \leq 10^9$) — количество рейсов, ожидающих отправления, и минимальный необходимый для очистки полосы зазор между временами отправления двух последовательных самолётов.

Во второй строке записана последовательность целых чисел t_1, t_2, \dots, t_n ($1 \leq t_1 < t_2 < \dots < t_n \leq 10^{18}$) — запланированные времена отправления рейсов.

Формат выходных данных

Выведите последовательность из n целых чисел, i -е из которых равняется реальному времени отправления рейса с номером i .

Пример

стандартный ввод	стандартный вывод
6 3 1 2 7 10 12 15	1 4 7 10 13 16

Замечание

Рассмотрим первый пример из условия:

1. Первый рейс вылетает в запланированное время.
2. Второй рейс будет задержан из-за необходимости уборки полосы.
3. Запланированной разницы по времени между третьим и вторым рейсом достаточно, чтобы третий рейс вылетел вовремя несмотря на задержку второго рейса и необходимость уборки полосы.
4. Четвёртый рейс вылетает вовремя.
5. Пятый рейс придётся задержать, так как между ним и четвёртым не был запланирован достаточный интервал по времени.
6. Изначально между пятым и шестым рейсом был запланирован интервал времени, достаточный для уборки взлётно-посадочной полосы, но, из-за задержки пятого рейса, шестой рейс так же вылетит с опозданием.

Система оценки

Тесты к этой задаче состоят из трёх групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **предыдущих** групп.

Группа	Баллы	Дополнительные ограничения		Комментарий
		n	k, t_i	
0	0	–	–	Тесты из условия.
1	50	$n \leq 100$	$k, t_i \leq 100$	
2	50	–	–	

Задача F. Проще не бывает!

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 6 секунд
Ограничение по памяти: 512 мегабайт

Вернувшись домой из школы, Иван долго думал о сегодняшнем занятии кружка по математике, на котором учитель рассказывал о бесконечных числовых последовательностях. В качестве одного из примеров рассматривалась следующая интересная последовательность целых положительных чисел:

$1, 1, 2, 1, 1, 2, 3, 2, 1, 1, 2, 3, 4, 3, 2, 1, \dots$

Учитель пояснил, что в этой последовательности каждое целое положительное число встречается бесконечное число раз. Однако Ивана заинтересовал ещё и другой вопрос: как определить, какое число находится в последовательности на месте под номером n ? На вопрос Ивана учитель ответил, что это очень просто, и предложил Ивану подумать над этой задачей самостоятельно.

Иван увлекается не только математикой, но и программированием, поэтому ему хочется реализовать алгоритм, который позволит быстро отвечать на поставленный вопрос для очень большого диапазона возможных n . Помогите ему в этом.

Формат входных данных

В первой строке входных данных находится одно целое число n ($1 \leq n \leq 10^{500\,000}$).

Формат выходных данных

Выведите одно целое число без пробелов и ведущих нулей — n -й элемент заданной последовательности.

Пример

стандартный ввод	стандартный вывод
7	3

Замечание

Жюри олимпиады не гарантирует существование решения, выполняющегося на всех тестах с двукратным запасом времени работы.

Система оценки

Тесты к этой задаче состоят из семнадцати групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **предыдущих** групп. Результаты проверки всех тестов доступны во время соревнования.

Группа	Баллы	Дополнительные ограничения	Комментарий
		n	
0	0	–	Тест из условия.
1	10	$n \leq 10^{18}$	
2	10	$n \leq 10^{100}$	
3	5	$n \leq 10^{1000}$	
4	5	$n \leq 10^{2000}$	
5	5	$n \leq 10^{5000}$	
6	5	$n \leq 10^{10\,000}$	
7	5	$n \leq 10^{20\,000}$	
8	5	$n \leq 10^{50\,000}$	
9	5	$n \leq 10^{75\,000}$	
10	5	$n \leq 10^{100\,000}$	
11	5	$n \leq 10^{150\,000}$	
12	5	$n \leq 10^{200\,000}$	
13	5	$n \leq 10^{250\,000}$	
14	5	$n \leq 10^{300\,000}$	
15	10	$n \leq 10^{400\,000}$	
16	10	–	

Задача G. Дорога на олимпиаду

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

Успешно решив все задачи отборочного этапа, Миша получил приглашение принять участие в очном туре Закрытой олимпиады школьников по информатике. Миша планирует добираться до места проведения олимпиады самолётами, при этом он хочет совершить не более одной пересадки, то есть использовать не более двух рейсов. Разумеется, среди всех таких вариантов его интересует самый дешёвый.

Жюри Закрытой олимпиады держит место и дату проведения очного этапа олимпиады в строжайшем секрете, а Миша очень любит путешествовать, поэтому не может заранее знать, из какого города он будет начинать свой путь. Кроме того, периодически появляются новые регулярные рейсы, а некоторые старые наоборот исчезают из расписания, поэтому Миша просит вас написать программу, которая будет отвечать на следующие запросы:

1. Добавить новый регулярный рейс.
2. Удалить некоторый рейс.
3. Определить минимальную стоимость маршрута между двумя городами, если рассматривать только маршруты без пересадок и маршруты с одной пересадкой.

Каждый рейс связывает некоторую пару городов u_i и v_i , и имеет свою стоимость c_i . Все рейсы **двусторонние**, то есть позволяют добраться как из города u_i в город v_i , так и наоборот, из города v_i в город u_i .

Формат входных данных

В первой строке входных данных записаны два целых числа n и m ($2 \leq n \leq 100\,000$, $0 \leq m \leq 100\,000$) — количество городов и количество уже доступных рейсов соответственно.

Следующие m строк описывают имеющиеся рейсы. В i -й из этих строк записаны три целых числа u_i , v_i и c_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$, $0 \leq c_i \leq 10^9$) — номера городов, которые связывает i -й регулярный рейс, и его стоимость.

В следующей строке записано одно целое число q ($1 \leq q \leq 100\,000$) — количество запросов.

Каждая из последующих q строк содержит описание запроса в одном из трёх форматов:

- $1\ u_i\ v_i\ c_i$ ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$, $0 \leq c_i \leq 10^9$) соответствует добавлению регулярного рейса между городами u_i и v_i стоимостью c_i .
- $2\ u_i\ v_i$ ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$) соответствует отмене регулярного рейса между городами u_i и v_i . Гарантируется, что в момент поступления такого запроса, города u_i и v_i соединены регулярным рейсом.
- $3\ u_i\ v_i$ ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$) означает, что Миша хочет знать самый дешёвый способ добраться из города u_i в город v_i , при условии использования не более, чем одной пересадки.

Гарантируется, что в любой момент времени между каждой парой городов существует не более одного регулярного рейса. Дополнительно гарантируется, что во входных данных присутствует хотя бы один запрос третьего типа.

Формат выходных данных

Для каждого запроса третьего типа выведите минимальную стоимость поездки с использованием не более одной пересадки. Если между указанной парой городов вовсе не существует подходящих маршрутов, то выведите -1 .

Пример

стандартный ввод	стандартный вывод
5 6	7
1 2 3	9
1 3 8	10
2 3 4	1
2 4 7	-1
3 4 1	1
4 5 7	4
13	14
3 1 3	-1
3 1 4	
2 1 3	
3 1 4	
1 1 3 0	
3 1 4	
3 1 5	
3 3 4	
1 1 4 1	
3 2 4	
3 2 5	
2 4 5	
3 2 5	

Замечание

Рассмотрим пример. Для запросов третьего типа перечислим оптимальные маршруты в порядке появления этих запросов во входных данных:

1. Через город 2 (хотя существует и прямой рейс, он не является наиболее выгодным маршрутом).
2. Через город 3.
3. Через город 2.
4. Через город 3.
5. Подходящего маршрута не существует.
6. Прямой рейс.
7. Через город 1.
8. Через город 4.
9. Подходящего маршрута не существует.

Система оценки

Тесты к этой задаче состоят из пяти групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов всех предыдущих групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Баллы	Доп. ограничения	Комментарий
		n, m, q	
0	0	–	Тесты из условия.
1	20	$n, m, q \leq 100$	
2	20	$n, m, q \leq 3000$	
3	20	–	Из города в любой момент летает не более 500 рейсов.
4	40	–	Offline-проверка.

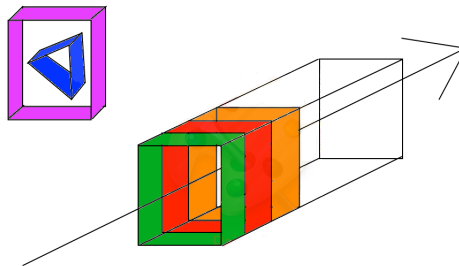
Задача Н. Макака и Бананы

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

Макака Виктор решил испытать удачу и отправился в широко известное в узких кругах «zino000», где он планирует опробовать новейший игровой автомат.

Автомат устроен очень просто: макака дергает за канат, чем приводит во вращение основной барабан (смотрите рисунок), внутри которого находятся n слоёв, расположенных перпендикулярно оси вращения барабана. В каждом слое находится жёстко закреплённый невырожденный треугольник. Таким образом, все треугольники вращаются вокруг оси барабана с одинаковой угловой скоростью. Внутри каждого треугольника расположен шарик пренебрежимо малого размера, который под действием гравитации скатывается в одну из вершин с минимальной текущей координатой y .

Ровно одна из вершин каждого треугольника является выигрышной. Если после остановки барабана шарик попадает в выигрышную вершину, то макака получает одну монету. При этом, если в момент остановки барабана у треугольника оказалось несколько вершин с минимальной координатой y , и одна из них является выигрышной, то Виктору обязательно повезёт и он получит монету.



После нескольких игр Виктор научился останавливать вращение автомата в произвольный момент времени с помощью ловкого пинка. Теперь он хочет знать, какое максимальное количество монет он может выиграть, если правильно выберет момент, когда прекратить вращение конструкции.

Формат входных данных

В первой строке записано единственное число n ($1 \leq n \leq 200\,000$) — количество треугольников в барабане.

В каждой из последующих i строк содержится шесть целых чисел $x_{1,i}, y_{1,i}, x_{2,i}, y_{2,i}, x_{3,i}, y_{3,i}$ — координаты точек треугольника в i -м слое барабана. Координаты даны в плоскости, ортогональной оси вращения барабана, при этом ось вращения проходит через точку $(0, 0)$. Выигрышной является первая вершина каждого из треугольников. Все координаты целые и по модулю не превосходят 10^9 . Гарантируется, что все треугольники являются невырожденными (то есть, площадь каждого треугольника строго больше нуля).

Формат выходных данных

В единственной строке выведите максимальное количество монет, которое может получить макака Виктор, если правильно выберет момент времени, когда число выигрышных вершин, обладающих минимальной y -координатой среди вершин своего треугольника, максимально.

Примеры

стандартный ввод	стандартный вывод
<pre>3 0 0 2 0 1 2 3 1 5 2 5 0 7 2 6 0 8 0</pre>	3
<pre>4 0 0 1 0 0 1 0 0 1 0 0 -1 0 0 -1 0 0 1 0 0 -1 0 0 -1</pre>	2

Замечание

Во втором примере одним из оптимальных решений будет остановить барабан в самом начале. Тогда выигрышная вершина окажется одной из нижних для треугольников на первом и третьем слое.

Система оценки

Тесты к этой задаче состоят из четырёх групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **предыдущих** групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

В таблице ниже через значение c обозначено максимально возможное абсолютное значение координат во вводе, то есть $-c \leq x_{1,i}, y_{1,i}, x_{2,i}, y_{2,i}, x_{3,i}, y_{3,i} \leq c$.

Подзадача	Баллы	Ограничения		Комментарий
		n	c	
0	0	—	—	Тесты из условия
1	30	$n \leq 5000$	$c = 20\,000$	—
2	30	$n \leq 200\,000$	$c = 20\,000$	—
3	40	$n \leq 200\,000$	$c = 10^9$	Offline-проверка.

Задача I. Разморозка таблицы

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	512 мегабайт

Только что завершился очный этап Открытой олимпиады школьников по программированию. В очном этапе принимало участие n школьников, для решения было предложено m задач (разумеется, интересных), за каждую задачу можно было получить любое количество баллов от 0 до k включительно. В некоторых задачах есть тесты с **offline-проверкой**, это означает, что результаты тестирования решения на данных тестах станут доступны только после окончания соревнования. В этот раз правила тестирования всех задач таковы, что решения будут выполняться на тестах с offline-проверкой только при прохождении всех обычных тестов (что, вообще говоря, не всегда верно для задач Открытой олимпиады).

Сейчас все школьники собрались в одном зале и ожидают начала церемонии закрытия. Публично доступна предварительная таблица результатов, в которой баллы всех школьников по всем задачам указаны без учёта баллов за тесты с offline-проверкой. При этом каждый из участников олимпиады знает свои итоговые баллы по всем задачам, то есть баллы с учётом тестов с offline-проверкой. Время от времени кто-либо из участников сообщает всем присутствующим свои баллы по какой-либо задаче, а некоторые из участников задаются вопросом, какое максимальное и какое минимальное место они могли занять с учётом предварительной таблицы и всей информации, сообщённой к данному моменту другими участниками.

Местом участника называется количество участников, набравших строго большее количество баллов чем он, увеличенное на один. Когда кто-либо из участников хочет выяснить своё максимальное и минимальное возможное итоговое место, он рассматривает все возможные итоговые таблицы, которые могли получиться с учётом 1) имеющейся предварительной таблицы, 2) информации, сообщённой другими участниками, и 3) правилами тестирования задач. Последнее означает, что баллы за offline-проверку можно получить, только набра полный балл по обычным тестам.

Формат входных данных

В первой строке записаны четыре целых числа n, m, q, k ($1 \leq n, m \leq 100\,000$, $1 \leq n \cdot m \leq 1\,000\,000$, $1 \leq q \leq 100\,000$, $1 \leq k \leq 10^9$) — количество участников олимпиады, количество задач, количество событий и максимальный балл за каждую задачу соответственно.

Во второй строке записаны m целых чисел s_1, s_2, \dots, s_m ($0 \leq s_i \leq k$), i -е из которых задаёт количество баллов за тесты с offline-проверкой в i -й задаче. Таким образом обычные тесты по i -й задаче стоят $k - s_i$ баллов.

Далее следуют n строк по m целых чисел в каждой. Через $a_{i,j}$ обозначим j -е число в i -й строке ($0 \leq a_{i,j} \leq k - s_j$) — количество баллов i -го участника по j -й задаче в предварительное таблице результатов.

Далее следуют q строк с описанием событий. Каждая строка начинается с целого числа t ($1 \leq t \leq 2$) — типа события.

Если $t = 1$, то далее следует одно целое число i ($1 \leq i \leq n$), которое означает, что участник с номером i хочет узнать своё максимальное и минимальное возможное место в итоговой таблице.

Если $t = 2$, то далее следуют три целых числа i, j, c ($1 \leq i \leq n$, $1 \leq j \leq m$, $a_{i,j} \leq c \leq k$), которые означают, что участник с номером i объявил всем, что его баллы по задаче j равны c .

Гарантируется, что никакой участник не называет баллы по одной и той же задаче дважды, и что участник называет свои баллы по задаче только при условии, что он набрал по ней максимально возможный балл без учёта тестов с offline-проверкой. Дополнительно гарантируется, что во входных данных присутствует хотя бы один запрос первого типа.

Формат выходных данных

Для каждого запроса первого типа выведите два целых числа — максимальное и минимальное место, которое мог занять данный участник, согласно имеющейся на текущий момент времени информации.

Примеры

стандартный ввод	стандартный вывод
2 2 2 100 0 0 100 100 100 100 1 1 1 2	1 1 1 1
2 2 2 100 10 0 90 100 89 100 1 1 1 2	1 1 2 2
2 2 2 100 10 0 90 100 90 100 1 1 1 2	1 2 1 2
2 2 6 100 10 0 90 100 90 100 1 1 1 2 2 1 1 90 2 2 1 100 1 1 1 2	1 2 1 2 2 2 1 1

Замечание

В первом тесте из условия оба участника делят первое место.

Во втором тесте из условия первый участник может набрать ещё 10 баллов, но даже если этого не произойдёт, он останется на первом месте. Баллы второго участника измениться не могут и он занимает второе место.

В третьем тесте из условия каждый из участников может набрать от 190 до 200 баллов и, в зависимости от этого, занять либо первое, либо второе место.

В четвёртом тесте из условия ситуация аналогична третьему тесту до тех пор, пока каждый из участников не объявит свои итоговые баллы. После этого места определяются однозначно.

Система оценки

Тесты к этой задаче состоят из трёх групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых предыдущих групп, указанных в таблице ниже. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Баллы	Дополнительные ограничения		Необх. группы	Комментарий
		n, m	t		
0	0	–	–	–	Тесты из условия
1	30	$1 \leq n, m \leq 1000$	–	0	–
2	30	–	$t = 1$	0	$t = 1$ для всех событий.
3	40	–	–	0,1,2	Offline-проверка.

Задача J. Дороги не роскошь, а место передвижения

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

Страна N состоит из n городов, пронумерованных от 1 до n , однако в ней совсем нет дорог! Президентом страны было принято решение построить m дорог в течение m дней, при этом каждый день будет строиться ровно одна новая дорога. Дополнительно, в целях поддержки миграции населения было принято решение сделать все дороги **односторонними**, и планировать дорожную сеть таким образом, что если существует путь из города i в город j , то пути из города j в город i **не существует**.

Для популяризации новых дорог среди населения страны N было решено ежедневно проводить серию автопробегов по городам страны. Каждая машина автопробега стартует в некотором городе, после чего совершает перемещения по уже построенным к данному дню дорогам и заканчивает свой маршрут также в каком-нибудь городе. При этом разрешается, чтобы город старта совпадал с городом финиша, то есть чтобы машина не совершала ни одного переезда по дорогам. Преследуя одновременно цели порадовать жителей страны и цели экономии топлива водителей, организаторы решили каждый день выбирать набор маршрутов согласно следующим правилам:

1. Каждый город должен быть посещён машиной автопробега. Посещёнными считаются начальный, конечный и все промежуточные города на пути любой из машин.
2. Никакой город не должен быть посещён машинами автопробега более одного раза.
3. Общее количество используемых машин должно быть минимальным.

Министерство транспорта страны N уже запланировало постройку ровно одной дороги для каждого из ближайших m дней. Теперь представители министерства просят вас для каждого из дней определить, какое минимальное количество машин потребуется для автопробега. Считайте, что в день i доступны дороги, построенные в дни $1, 2, \dots, i$.

Формат входных данных

В первой строке входных данных записаны два числа n и m ($2 \leq n \leq 1000$, $1 \leq m \leq 100\,000$) — количество городов в стране N и количество дней, в течение которых будут строиться дороги.

В каждой из последующих m строк записаны два целых числа u_i и v_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$), означающих что в i -й день будет построена односторонняя дорога из города u_i в город v_i . Допускается, что будет построена дорога между двумя городами, уже соединёнными дорогой ранее. Гарантируется, что после постройки любой дороги, если существует путь из города a в город b , то не существует пути из города b в город a .

Формат выходных данных

Выведите m чисел, i -е из которых равняется минимальному количеству машин для автопробега после постройки первых i дорог.

Примеры

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

Замечание

В первом примере после построения первых двух дорог всё равно потребуется запустить как минимум две машины. Например, одна из них может проехать из города 1 в город 2, а другая начнёт в городе 3 и закончит автопробег в нём же. После построения всех дорог достаточно одной машины, которая поедет по маршруту $1 \rightarrow 2 \rightarrow 3$.

Во втором примере, после построения всех дорог можно было бы посетить все города используя только два автомобиля, но этому мешает условие, что никакой город не должен быть посещён больше, чем одним автомобилем пробега.

Система оценки

Тесты к этой задаче состоят из четырёх групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых предыдущих групп, указанных в таблице ниже. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Баллы	Дополнительные ограничения		Комментарий
		n	m	
0	0	–	–	Тесты из условия.
1	30	$n \leq 100$	$m \leq 100$	
2	30	–	$m \leq 10\,000$	
3	40	–	–	Offline-проверка.