

Задача 1. Кампус

Имя входного файла: `building.in`
Имя выходного файла: `building.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Новое здание кампуса Университета Байтбурга имеет n этажей, пронумерованных снизу вверх от 1 до n . Комнаты студентов расположены в нескольких подъездах.

В каждом подъезде на этажах, номер которых кратен числу k , расположено по x комнат, а на остальных этажах расположено по y комнат.

Комнаты внутри каждого подъезда пронумерованы последовательными натуральными числами. Номера комнат на первом этаже имеют наименьшие значения в этом подъезде, затем следуют номера комнат на втором этаже, и так далее. Комнаты в первом подъезде пронумерованы, начиная с 1, в каждом следующем подъезде нумерация комнат начинается с числа, следующего после максимального номера комнаты в предыдущем подъезде.

На рис. 1 показаны номера комнат в здании с $n = 7$ этажами, 3 подъездами, и параметрами $k = 3$, $x = 2$, $y = 3$.

	Подъезд 1	Подъезд 2	Подъезд 3
7 этаж	17, 18, 19	36, 37, 38	55, 56, 57
6 этаж	15, 16	34, 35	53, 54
5 этаж	12, 13, 14	31, 32, 33	50, 51, 52
4 этаж	9, 10, 11	28, 29, 30	47, 48, 49
3 этаж	7, 8	26, 27	45, 46
2 этаж	4, 5, 6	23, 24, 25	42, 43, 44
1 этаж	1, 2, 3	20, 21, 22	39, 40, 41

Рис. 1. Пример нумерации комнат в здании

Для организации расселения студентов администрация кампуса должна по номеру комнаты оперативно определять этаж, на котором она находится.

Требуется написать программу, которая по заданным числам n , k , x и y , а также по номерам комнат, определяет для каждой комнаты, на каком этаже она находится.

Формат входного файла

Первая строка входного файла содержит натуральные числа n , k , x и y ($1 \leq n \leq 10^9$, $1 \leq k \leq n$, $1 \leq x, y \leq 10^9$). Соседние числа разделены ровно одним пробелом.

Вторая строка входного файла содержит натуральное число q — количество номеров комнат, для которых требуется определить этаж ($1 \leq q \leq 1000$).

Третья строка содержит q целых чисел a_1, a_2, \dots, a_q — номера комнат ($1 \leq a_i \leq 10^{18}$). Можно считать, что в здании так много подъездов, что все комнаты с заданными номерами существуют.

Формат выходного файла

Требуется вывести q чисел, по одному на строке. Для каждого номера комнаты во входном файле требуется вывести номер этажа, на котором она находится.

Пример входных и выходных файлов

building.in	building.out
7 3 2 3	1
4	7
1 19 20 50	1
	5

Описание подзадач и системы оценивания

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

Подзадача	Баллы	Дополнительные ограничения		Необходимые подзадачи
		n, x, y	q, a_i	
1	31	$1 \leq n \leq 10,$ $1 \leq x, y \leq 10$	$q = 1$ $1 \leq a_i \leq 100$	
2	19	$1 \leq n \leq 10^7,$ $1 \leq x, y \leq 10^9$	$q = 1$ $1 \leq a_i \leq 10^7$	1
3	16	$1 \leq n \leq 10^9,$ $1 \leq x, y \leq 10^9,$ $x = y$	$1 \leq q \leq 1000$ $1 \leq a_i \leq 10^{18}$	
4	34	$1 \leq n \leq 10^9$ $1 \leq x, y \leq 10^9$	$1 \leq q \leq 1000$ $1 \leq a_i \leq 10^{18}$	1, 2, 3

Получение информации о результатах окончательной проверки

По запросу сообщается результат окончательной проверки на каждом тесте.

Задача 2. Калькулятор

Имя входного файла: `calc.in`
Имя выходного файла: `calc.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

В качестве домашнего задания по информатике ученикам предложено разработать специальный калькулятор, который устроен следующим образом.

Сначала пользователь вводит целое положительное число n , которое выводится на экран. Затем пользователь может нажимать на три кнопки: А, В и С.

При нажатии на кнопку А число, которое выведено на экран, делится на 2. Если число на экране нечетное, то остаток отбрасывается. Например, результат этой операции для числа 80 равен 40, а для числа 239 равен 119.

При нажатии на кнопку В к числу, которое выведено на экран, прибавляется 1, и результат делится на 2. Остаток от деления отбрасывается. Например, результат операции для числа 80 равен 40, а для числа 239 равен 120.

При нажатии на кнопку С происходит следующее. Если число, которое выведено на экран, положительное, то из него вычитается 1 и результат делится на 2, остаток отбрасывается. Если же перед нажатием на кнопку С на экран было выведено число 0, то оно остается неизменным. Например, результат операции для числа 80 равен 39, а для числа 239 равен 119.

Пользователь ввел число n и собирается нажать на кнопки операций в некотором порядке. В частности, он планирует нажать на кнопку А суммарно a раз, на кнопку В – b раз и на кнопку С – c раз. Его заинтересовал вопрос, какое минимальное число может получиться в результате выполнения описанных операций.

Требуется написать программу, которая по введенному числу n и числам a , b и c , показывающим количество произведенных на калькуляторе операций разного типа, определяет минимальное число, которое может получиться в результате работы калькулятора.

Формат входного файла

Входной файл содержит четыре целых числа: n , a , b и c ($1 \leq n \leq 10^{18}$, $0 \leq a, b, c \leq 60$). Числа заданы на одной строке, соседние числа разделены одним пробелом.

Формат выходного файла

Требуется вывести одно число — минимальное число, которое может получиться у пользователя в результате работы калькулятора.

Примеры входных и выходных файлов

<code>calc.in</code>	<code>calc.out</code>
72 2 1 1	4

Пояснение к примеру

В примере пользователю необходимо оптимально действовать следующим образом: нажать на кнопку С и получить число 36, затем нажать на кнопку А и получить число 18, затем нажать на кнопку В и получить число 8, затем второй раз нажать на кнопку А и получить число 4.

Описание подзадач и системы оценивания

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

Подзадача	Баллы	Дополнительные ограничения		Необходимые подзадачи
		n	Дополнительные условия на a, b, c	
1	26	$1 \leq n \leq 10^9$	$0 \leq (a + b + c) \leq 7$	
2	23	$1 \leq n \leq 10^{18}$	$c = 0$	
3	24	$1 \leq n \leq 10^{18}$	$b = 0$	
4	27	$1 \leq n \leq 10^{18}$		1, 2, 3

Получение информации о результатах окончательной проверки

По запросу сообщается результат окончательной проверки на каждом тесте.

Задача 3. Размещение данных

Имя входного файла: data.in
Имя выходного файла: data.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Телекоммуникационная сеть крупной IT-компании содержит n серверов, пронумерованных от 1 до n . Некоторые пары серверов соединены двусторонними каналами связи, всего в сети m каналов. Гарантируется, что сеть серверов устроена таким образом, что по каналам связи можно передавать данные с любого сервера на любой другой сервер, возможно с использованием одного или нескольких промежуточных серверов.

Множество серверов A называется *отказоустойчивым*, если при недоступности любого канала связи выполнено следующее условие. Для любого не входящего в это множество сервера X существует способ передать данные по остальным каналам на сервер X хотя бы от одного сервера из множества A .

На рис. 1 показан пример сети и отказоустойчивого множества из серверов с номерами 1 и 4. Данные на сервер 2 можно передать следующим образом. При недоступности канала между серверами 1 и 2 — с сервера 4, при недоступности канала между серверами 2 и 3 — с сервера 1. На серверы 3 и 5 при недоступности любого канала связи можно по другим каналам передать данные с сервера 4.

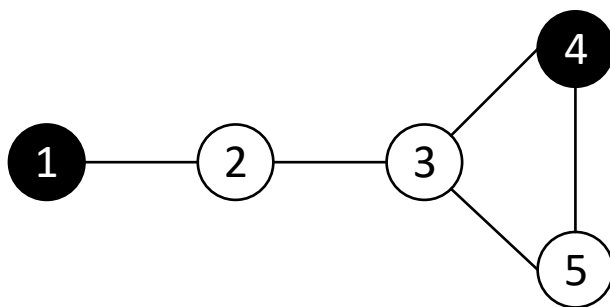


Рис. 1. Пример сети и отказоустойчивого множества серверов.

В рамках проекта группе разработчиков компании необходимо разместить свои данные в сети. Для повышения доступности данных и устойчивости к авариям разработчики хотят продублировать свои данные, разместив их одновременно на нескольких серверах, образующих отказоустойчивое множество. Чтобы минимизировать издержки, необходимо выбрать минимальное по количеству серверов отказоустойчивое множество. Кроме того, чтобы узнать, насколько гибко устроена сеть, необходимо подсчитать количество способов выбора такого множества, и поскольку это количество способов может быть большим, необходимо найти остаток от деления этого количества способов на число $10^9 + 7$.

Требуется написать программу, которая по заданному описанию сети определяет следующие числа: k — минимальное количество серверов в отказоустойчивом множестве серверов, s — остаток от деления количества способов выбора отказоустойчивого множества из k серверов на число $10^9 + 7$

Формат входного файла

Первая строка входного файла содержит целые числа n и m — количество серверов и количество каналов связи соответственно ($2 \leq n \leq 200\,000$, $1 \leq m \leq 200\,000$).

Следующие m строк содержат по два целых числа и описывают каналы связи между серверами. Каждый канал связи задается двумя целыми числами: номерами серверов, которые он соединяет.

Гарантируется, что любые два сервера соединены напрямую не более чем одним каналом связи, никакой канал не соединяет сервер сам с собой, и для любой пары серверов существует способ передачи данных с одного из них на другой, возможно с использованием одного или нескольких промежуточных серверов.

Формат выходного файла

Выведите два целых числа, разделенных пробелом: k — минимальное число серверов в отказоустойчивом множестве серверов, s — количество способов выбора отказоустойчивого множества из k серверов, взятое по модулю $10^9 + 7$

Пример входных и выходных файлов

data.in	data.out
5 5 1 2 2 3 3 4 3 5 4 5	2 3

Пояснение к примеру

В приведенном примере отказоустойчивыми являются следующие множества из двух серверов: $\{1, 3\}$, $\{1, 4\}$, $\{1, 5\}$.

Описание подзадач и системы оценивания

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

Подзадача	Баллы	Дополнительные ограничения		Необходимые подзадачи
		N	m	
1	25	$2 \leq n \leq 10$	$1 \leq m \leq 45$	
2	27	$2 \leq n \leq 200\,000$	$m = n - 1$	
3	28	$2 \leq n \leq 1000$	$1 \leq m \leq 5000$	1
4	21	$2 \leq n \leq 200\,000$	$1 \leq m \leq 200\,000$	1, 2, 3

Получение информации о результатах окончательной проверки

По запросу сообщается результат окончательной проверки на каждом тесте.

Задача 4. Полезные ископаемые

Имя входного файла: mining.in
Имя выходного файла: mining.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Ведется проект по освоению планеты соседней звездной системы. Для добычи полезных ископаемых планируется направить на планету несколько партий роботов.

Участок поверхности планеты, на котором планируется добывать полезные ископаемые, представляет собой клетчатый прямоугольник размером w на h , клетки участка имеют координаты от $(1, 1)$ до (w, h) . В некоторых клетках участка находятся базы специалистов, в которые могут быть доставлены партии роботов. Всего на участке размещено s баз, и i -я база находится в клетке с координатами (x_i, y_i) .

Каждая партия роботов характеризуется тремя параметрами: j -я партия доставляется на базу b_j , содержит n_j роботов и каждый робот партии обладает мобильностью m_j .

Когда партия роботов доставляется на соответствующую базу, каждый робот этой партии перемещается по поверхности планеты от базы до некоторой клетки. Если мобильность робота равна m , он может не более m раз переместиться на одну из восьми соседних клеток, как показано на рис. 1.

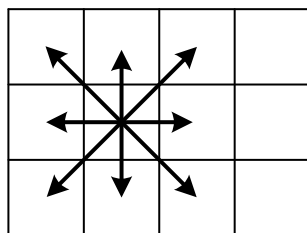


Рис. 1. Возможные перемещения робота в восьми направлениях.

После того как роботы из всех доставленных партий размещаются на участке, они активируются и начинают добычу полезных ископаемых. В процессе перемещения в одной клетке может одновременно находиться произвольное количество роботов. Однако после активации в каждой клетке должно находиться не более q роботов.

Руководством проекта получена информация о t партиях роботов, которые могут быть последовательно отправлены на планету. После доставки всех партий роботов, учитывая их ограниченную мобильность, возможна ситуация, что не удастся разместить роботов на участке так, чтобы в каждой клетке оказалось не больше q роботов. Поэтому руководство должно выбрать k первых партий роботов, где $0 \leq k \leq t$, которые будут полностью доставлены на соответствующие базы. После этого, если $k < t$, следует дополнительно принять z из n_{k+1} роботов следующей, $(k+1)$ -й партии, $0 \leq z < n_{k+1}$.

Все полученные таким образом роботы должны с учетом ограничения на мобильность разместиться на участке таким образом, чтобы в каждой клетке было не более q роботов. После этого они будут активированы и начнут добычу полезных ископаемых. Разумеется, руководство проекта старается максимизировать количество роботов, которые будут доставлены на планету, поэтому, с учетом описанных ограничений, требуется максимизировать k , а затем максимизировать z .

Требуется написать программу, которая по размерам участка, числу q , описанию расположения баз, а также количеству запланированных партий роботов и их описанию определяет максимальное число k — количество партий роботов, и затем — максимальное число z — дополнительное количество роботов из $(k+1)$ -й партии, чтобы, доставив роботов на планету, их можно было разместить на участке таким образом, чтобы в каждой клетке оказалось не более q роботов.

Формат входного файла

Первая строка входного файла содержит числа w , h , s и q ($1 \leq w, h \leq 10^5$, $1 \leq s \leq 4$, $1 \leq q \leq 100$). Последующие s строк содержат по два целых числа x_i , y_i и описывают базы специалистов ($1 \leq x_i \leq w$, $1 \leq y_i \leq h$).

Следующая строка содержит число t — количество партий роботов ($1 \leq t \leq 100$). Последующие t строк описывают партии роботов и содержат по 3 целых числа: b_j , n_j и m_j ($1 \leq b_j \leq s$, $1 \leq n_j \leq w \cdot h \cdot q$, $0 \leq m_j < \max(w, h)$).

Формат выходного файла

Требуется вывести два числа: k и z , $0 \leq k \leq t$. Если $k = t$, то z должно быть равно 0, иначе должно выполняться условие $0 \leq z < n_{k+1}$.

Пример входных и выходных файлов

mining.in	mining.out
4 3 2 1 1 1 3 2 3 1 4 1 2 9 1 1 12 2	1 7

Пояснение к примеру

В приведенном примере описания входных данных следует полностью принять первую партию роботов и дополнительно принять 7 роботов из второй партии. На рис. 2 показано, как можно разместить этих роботов на участке, чтобы в каждой клетке было не более одного робота. Базы специалистов показаны кружками. Клетки, в которых окажутся роботы с базы 1, показаны вертикальной штриховкой, а клетки, в которых окажутся роботы с базы 2, показаны серым цветом.

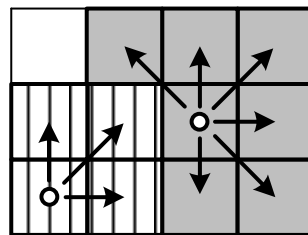


Рис. 2. Возможное размещение роботов на участке в данном примере.

Описание подзадач и системы оценивания

Баллы за каждую из подзадач 1–5 начисляются только в случае, если все тесты этой подзадачи и необходимых подзадач успешно пройдены.

Тесты для подзадачи 6 запускаются только в случае, если все тесты подзадач 1–5 успешно пройдены. Каждый тест в подзадаче 6 оценивается независимо в 1 балл.

Подзадача	Баллы	Дополнительные ограничения			Необходимые подзадачи
		w, h	s	q	
1	18	$1 \leq w, h \leq 20$	$s = 1$	$q = 1$	
2	12	$1 \leq w, h \leq 20$	$1 \leq s \leq 2$	$q = 1$	1
3	9	$1 \leq w, h \leq 20$	$1 \leq s \leq 3$	$q = 1$	1, 2
4	10	$1 \leq w, h \leq 20$	$1 \leq s \leq 3$	$1 \leq q \leq 100$	1, 2, 3
5	15	$1 \leq w, h \leq 10^5$	$s = 1$	$1 \leq q \leq 100$	1
6	36	$1 \leq w, h \leq 10^5$	$1 \leq s \leq 4$	$1 \leq q \leq 100$	1, 2, 3, 4, 5

Получение информации о результатах окончательной проверки

По запросу сообщаются баллы за каждую подзадачу.

Задача 5. Автоматизированное управление доставкой

Имя входного файла: `delivery.in`
Имя выходного файла: `delivery.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Группа программистов регионального сортировочного центра работает над автоматизацией управления доставкой почты.

Посылки принимаются в клиентских почтовых пунктах. Почтовый пункт принимает посылки, вес каждой из которых составляет целое число килограммов. Минимальный вес посылки равен 1 кг, а максимальный вес — k кг. Принятые посылки помещаются в специальный пакет.

Если после приема очередной посылки суммарный вес посылок в пакете больше или равен x кг, то пакет доставляется в муниципальный почтовый центр, где пакет с посылками перемещается в специальный контейнер.

Если после доставки очередного пакета суммарный вес посылок в контейнере больше или равен y кг, то контейнер перевозится в региональный сортировочный центр, откуда посылки уже доставляются получателям.

Суммарный вес посылок в контейнере при его перевозке может различаться в зависимости от массы принятых посылок. Необходимо выяснить, каким может быть минимальный суммарный вес посылок в контейнере при перевозке его из муниципального почтового центра в региональный сортировочный центр.

Требуется написать программу, которая по заданным значениям k — максимального веса посылки, x — необходимого веса пакета для его отправки в муниципальный почтовый центр, и y — необходимого веса контейнера для его отправки в региональный сортировочный центр, определяет минимальный вес контейнера при его перевозке.

Формат входного файла

Входной файл содержит три целых положительных числа, по одному на строке. Первая строка содержит число k ($1 \leq k \leq 10^9$). Вторая строка содержит число x ($1 \leq x \leq 10^9$). Третья строка содержит число y ($1 \leq y \leq 10^9$).

Формат выходного файла

Требуется вывести одно целое число — минимальный возможный вес контейнера при перевозке.

Пример входных и выходных файлов

<code>delivery.in</code>	<code>delivery.out</code>
2	21
7	
20	

Пояснение к примеру

В приведенном примере принимаются посылки весом 1 и 2 кг. При накоплении посылок с суммарным весом хотя бы в 7 кг пакет доставляется из клиентского почтового пункта в муниципальный почтовый центр. При накоплении посылок с суммарным весом хотя бы в 20 кг контейнер перевозится из муниципального почтового центра в региональный сортировочный центр.

Минимальный возможный вес контейнера в данном примере составляет 21 кг и достигается, например, следующим образом: в муниципальный почтовый центр

последовательно доставляется 3 пакета по 7 кг каждый. Пакет весом 7 кг может получиться, например, после приема семи посылок по 1 кг.

Описание подзадач и системы оценивания

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

Подзадача	Баллы	Ограничения		Необходимые подзадачи
		k	x, y	
1	21	$k = 1$	$1 \leq x, y \leq 100$	
2	18	$k = 2$	$1 \leq x, y \leq 100$	
3	21	$1 \leq k \leq 100$	$1 \leq x, y \leq 100$	1, 2
4	17	$1 \leq k \leq 40\,000$	$1 \leq x, y \leq 40\,000$	1, 2, 3
5	23	$1 \leq k \leq 10^9$	$1 \leq x, y \leq 10^9$	1, 2, 3, 4

Получение информации о результатах окончательной проверки

По запросу сообщается результат окончательной проверки на каждом тесте.

Задача 6. Большой линейный коллайдер

Имя входного файла: linear.in
Имя выходного файла: linear.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Группа ученых работает в международной научной лаборатории, которая занимается исследованиями поведения элементарных частиц в установке для экспериментов «Большой линейный коллайдер» (БЛК). Установка БЛК представляет собой прямую, в некоторых точках которой размещаются частицы, которые могут перемещаться вдоль прямой.

В очередном эксперименте в БЛК размещаются n частиц, каждая из которых представляет собой либо отрицательно заряженную частицу — электрон e^- , либо положительно заряженную частицу — позитрон e^+ . В эксперименте i -я частица исходно размещается в точке с координатой x_i . После начала эксперимента в результате работы БЛК частицы начнут перемещаться в разные стороны вдоль прямой: e^- частицы перемещаются по направлению уменьшения координаты, а e^+ частицы — по направлению увеличения координаты. Абсолютные величины скоростей всех частиц одинаковы и равны 1.

Если в процессе перемещения частицы e^- и e^+ оказываются в одной точке, то они взаимодействуют и обе исчезают, при этом они не влияют на дальнейшее поведение остальных частиц.

Ученые выбрали m различных моментов времени t_1, t_2, \dots, t_m , для каждого из которых их интересует, какое количество частиц находится в БЛК непосредственно после каждого из этих моментов времени. Отсчет времени начинается с момента 0, когда частицы приходят в движение. Частицы, исчезнувшие в результате взаимодействия в момент времени t_j , не должны учитываться при подсчете количества частиц для этого момента времени.

Требуется написать программу, которая по описанию исходного расположения и типов частиц, а также заданным моментам времени, определяет для каждого из моментов количество частиц, которое будет находиться в БЛК непосредственно после этого момента.

Формат входного файла

Первая строка входного файла содержит число n — количество частиц ($1 \leq n \leq 200\,000$). Последующие n строк описывают частицы следующим образом: каждая строка содержит по два целых числа x_i и v_i — координату i -й частицы и ее тип соответственно ($-10^9 \leq x_1 < x_2 < \dots < x_n \leq 10^9$, v_i равно -1 или 1). Частица e^- описывается значением $v_i = -1$, а частица e^+ описывается значением $v_i = 1$.

Следующая строка содержит целое число m — количество моментов времени, которые выбрали ученые ($1 \leq m \leq 200\,000$). Последняя строка содержит m целых чисел: t_1, t_2, \dots, t_m ($0 \leq t_1 < t_2 < \dots < t_m \leq 10^9$).

Формат выходного файла

Для каждого момента времени во входном файле требуется вывести одно число: количество частиц в БЛК непосредственно после этого момента.

Примеры входных и выходных файлов

<code>linear.in</code>	<code>linear.out</code>
4 -1 1 0 -1 1 1 5 -1 4 0 1 2 3	4 2 0 0

Пояснение к примеру

В приведенном примере в начальный момент в БЛК находятся 4 частицы: частица e^+ в точке -1 , частица e^- в точке 0 , частица e^+ в точке 1 и частица e^- в точке 5 .

В момент времени 0.5 первая частица e^+ и первая частица e^- сталкиваются в точке с координатой -0.5 и исчезают. В момент времени 1 оставшиеся две частицы находятся в точках с координатами 2 и 4 , соответственно. В момент времени 2 они сталкиваются в точке 3 и исчезают. Больше в БЛК частиц нет.

Описание подзадач и системы оценивания

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

Под-задача	Баллы	Дополнительные ограничения				Необх. подзадачи
		n	x_i	m	t_i	
1	35	$1 \leq n \leq 100$	$-100 \leq x_i \leq 100$	$m = 1$	$0 \leq t_1 \leq 100$	
2	12	$1 \leq n \leq 100$	$-10^9 \leq x_i \leq 10^9$	$m = 1$	$0 \leq t_1 \leq 10^9$	1
3	12	$1 \leq n \leq 200\,000$	$-10^9 \leq x_i \leq 10^9$	$m = 1$	$0 \leq t_1 \leq 10^9$	1, 2
4	41	$1 \leq n \leq 200\,000$	$-10^9 \leq x_i \leq 10^9$	$1 \leq m \leq 200\,000$	$0 \leq t_1 \leq 10^9$	1, 2, 3

Получение информации о результатах окончательной проверки

По запросу сообщается результат окончательной проверки на каждом тесте.

Задача 7. Силовые поля

Имя входного файла: power.in
Имя выходного файла: power.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

В физико-биологической лаборатории исследуют воздействие излучения на растения при облучении через *силовые поля*.

Экспериментальная установка содержит квадратную платформу размером $10^9 \times 10^9$, заполненную плодородной почвой. Над платформой установлен источник излучения. Между источником излучения и платформой можно включать n силовых полей.

Генератор силовых полей установлен над точкой $(0, 0)$. При этом i -е силовое поле представляет собой прямоугольник со сторонами, параллельными границам платформы и координатами двух противоположных углов $(0, 0)$ и (x_i, y_i) .

В эксперименте планируется изучать воздействие излучения на растения при облучении через k силовых полей. Из заданных n полей необходимо выбрать k полей для эксперимента. Ученые хотят выбрать силовые поля таким образом, чтобы площадь участка платформы, над которой находятся все k выбранных силовых полей, была максимальна.

Требуется написать программу, которая по заданным целым числам n , k и описанию n силовых полей определяет, какие k силовых полей необходимо выбрать для эксперимента, чтобы площадь участка, покрытого всеми k силовыми полями, была максимальна, и выводит площадь этого участка.

Формат входного файла

Первая строка входного файла содержит целые числа n и k ($1 \leq k \leq n \leq 200\,000$) — общее количество силовых полей и количество силовых полей, которые необходимо выбрать для эксперимента.

Последующие n строк содержат по два целых числа x_i , y_i ($1 \leq x_i, y_i \leq 10^9$) — координаты дальнего от начала координат угла прямоугольного участка i -го силового поля.

Формат выходного файла

Требуется вывести одно целое число: максимальную площадь искомого участка

Пример входных и выходных файлов

power.in	power.out
5 3 3 5 2 2 2 5 4 4 5 3	9

Пояснение к примеру

На рис. 1 показаны пять силовых полей, заданных во входном файле. Оптимальный способ выбрать из них три поля для эксперимента показан на рис. 2.

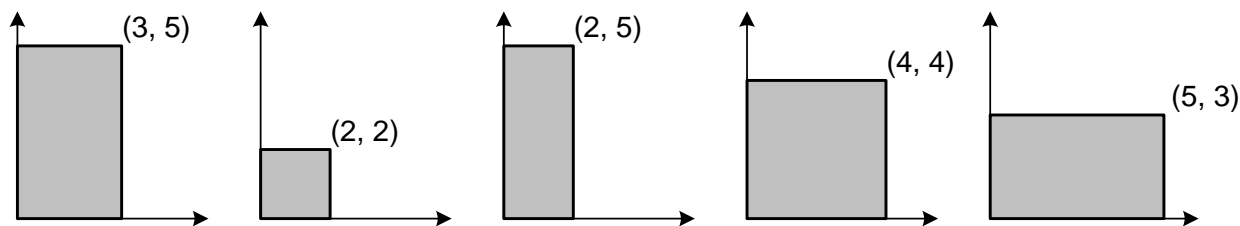


Рис 1. Силовые поля в примере описания входных данных.

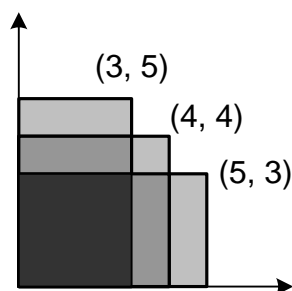


Рис 2. Оптимальный выбор трех из пяти силовых полей в данном примере.

Описание подзадач и системы оценивания

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

Подзадача	Баллы	Ограничения		Необходимые подзадачи
		n	k	
1	18	$1 \leq n \leq 20$	$1 \leq k \leq n$	
2	25	$1 \leq n \leq 300$	$1 \leq k \leq n$	1
3	20	$1 \leq n \leq 3\,000$	$1 \leq k \leq n$	1, 2
4	17	$2 \leq n \leq 200\,000$	$k = 2$	
5	20	$1 \leq n \leq 200\,000$	$1 \leq k \leq n$	1, 2, 3, 4

Получение информации о результатах окончательной проверки

По запросу сообщается результат окончательной проверки на каждом тесте.

Задача 8. Повышение квалификации

Имя входного файла: `qual.in`
Имя выходного файла: `qual.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Взаимодействие сотрудников в некоторой компании организовано в виде иерархической структуры. Всего в компании работают n сотрудников. Каждому сотруднику присвоен уникальный номер от 1 до n , директору присвоен номер 1. У каждого сотрудника, кроме директора, есть ровно один непосредственный начальник. Непосредственный начальник сотрудника i имеет номер p_i , причем $p_i < i$.

Сотрудник x является подчиненным уровня 1 сотрудника y , если $p_x = y$. Для $k > 1$ сотрудник x является подчиненным уровня k сотрудника y , если сотрудник p_x является подчиненным уровня $k - 1$ сотрудника y .

У директора компании появилась возможность направить некоторых сотрудников на курсы повышения квалификации. Для этого он решил выбрать два числа L и R и направить на курсы всех сотрудников с номерами i , такими что $L \leq i \leq R$.

Перед тем, как выбрать числа L и R , директор получил m пожеланий от сотрудников компании, j -е пожелание задается двумя числами u_j и k_j и означает, что сотрудник u_j просит отправить на курсы одного из своих подчиненных уровня k_j . Для экономии средств директор хочет выбрать такие L и R , чтобы количество сотрудников, направленных на повышение квалификации, было минимальным возможным, но при этом все пожелания были выполнены.

Требуется написать программу, которая по заданным в компании отношениям начальник-подчиненный и пожеланиям сотрудников определяет такие числа L и R , что если отправить на курсы повышения квалификации всех сотрудников с номерами от L до R включительно, то все пожелания будут выполнены, а количество сотрудников, направленных на повышение квалификации, будет минимальным возможным. Если оптимальных пар чисел L, R будет несколько, требуется найти ту из них, в которой значение L минимально.

Формат входного файла

Первая строка входного файла содержит число n — количество сотрудников компании ($2 \leq n \leq 200\,000$). Вторая строка содержит $(n - 1)$ чисел: p_2, p_3, \dots, p_n ($1 \leq p_i < i$) — номера непосредственных начальников сотрудников.

Третья строка содержит число m — количество пожеланий от сотрудников.

Последующие m строк задают пожелания сотрудников и содержат по два целых числа u_j, k_j ($1 \leq u_j < n, 1 \leq k_j < n$, гарантируется, что у сотрудника u_j есть хотя бы один подчиненный уровня k_j).

Формат выходного файла

Необходимо вывести два искоемых числа: L и R . Если оптимальных пар (L, R) несколько, требуется вывести ту, в которой значение L минимально.

Пример входных и выходных файлов

qual.in	qual.out
7 1 1 2 2 3 3 3 1 1 3 1 1 2	3 6

Пояснение к примеру

На повышение квалификации будут направлены сотрудники с номерами 3, 4, 5 и 6. Сотрудник с номером 3 является подчиненным уровня 1 сотрудника с номером 1, сотрудник с номером 4 — подчиненным уровня 2 сотрудника с номером 1, а сотрудник с номером 6 — подчиненным уровня 1 сотрудника с номером 3.

Описание подзадач и системы оценивания

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

Подзадача	Баллы	Ограничения		Необходимые подзадачи
		n	Дополнительные условия	
1	19	$2 \leq n \leq 50$		
2	25	$2 \leq n \leq 3000$		1
3	21	$2 \leq n \leq 200\,000$	для всех i выполнено $p_i = i - 1$	
4	35	$2 \leq n \leq 200\,000$		1, 2, 3

Получение информации о результатах окончательной проверки

По запросу сообщаются баллы за каждую подзадачу.