

## Задача 5. Челябинск

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	64 мегабайта

Все знают, что челябинцы — чемпионы в соревнованиях по слепому парному программированию. И чтобы доказать это окончательно, было решено провести Всероссийскую Парную Олимпиаду Школьников по Программированию Слепую.

У каждого потенциального участника олимпиады имеется его навык — натуральное число  $a_i$ . Чтобы олимпиада прошла честно, должен быть способ разделить  $N$  участников олимпиады на  $\frac{N}{2}$  команды из двух человек так, чтобы сумма навыков участников во всех командах была одинаковая (это означает, что  $N$  чётное). Назовем такой набор участников *идеально сбалансированным*.

Челябинская сборная состоит из  $N$  участников, каждый со своим уникальным навыком, а поскольку они чемпионы, их сборная представляет собой *идеально сбалансированный* набор. В олимпиаде, помимо представителей Челябинска, участвуют и другие школьники в количестве  $K$  человек. Их навыки программирования также уникальны.

Теперь вы должны выбрать из этих  $N + K$  человек ровно  $N$  участников олимпиады. Про каждого участника вы знаете число  $a_i$  — его навык, но не знаете какой город он представляет. Конечно, участники олимпиады должны быть *идеально сбалансированным* набором.

### Формат входных данных

В первой строке находится два целых числа  $N$  и  $K$  ( $1 \leq N \leq 150,000$ ;  $1 \leq K \leq 400$ ). На следующей строке находятся  $N + K$  натуральных различных чисел  $a_i$  в возрастающем порядке ( $1 \leq a_i \leq 10^9$ ).

### Формат выходных данных

Выведите в одной строке  $N$  натуральных различных чисел  $a_i$  в возрастающем порядке. Ваш вывод должен составлять подмножество навыков из входных данных и образовывать *идеально сбалансированный* набор участников.

### Система оценки

Подзадача	Баллы	Ограничения	Необходимые подзадачи	Результаты
1	11	$N \leq 2000, K = 1$		Первая ошибка
2	9	$K = 1$	1	Первая ошибка
3	14	$K = 2$		Первая ошибка
4	15	$N, K \leq 100$		Первая ошибка
5	9	$N + K \leq 18$		Первая ошибка
6	14	$N \leq 2000, K \leq 20$	1, 5	Первая ошибка
7	15	$K \leq 20$	1–3, 5, 6	Первая ошибка
8	13		1–7	Только баллы

## Примеры

стандартный ввод	стандартный вывод
2 1 1 2 3	1 2
4 3 1 3 4 5 6 7 8	1 3 4 6

## Задача 6. Липецк

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

Мало кто знает, но в 1974 году в Липецке был произведен единственный советский игровой автомат с игрой «Бешеный кролик». Игровое поле представляет собой  $n$  горизонтальных этажей. Каждый этаж представляет последовательность из  $m$  плит одинаковой длины. Некоторые из этих плит являются люками, которые игрок может открывать с помощью кнопки на автомате. Другие плиты неподвижны, и никак не меняются в процессе игры. Каждый этаж располагается ровно друг под другом, а сами этажи нумеруются с единицы **сверху вниз**. Таким образом  $j$ -я плита на этаже с номером  $i$  находится ровно над  $j$ -й плитой на этаже с номером  $i + 1$ .

Игровой процесс заключается в следующем: в начале уровня в определенном месте появляется кролик, который начинает бешено бежать направо. Одновременно с появлением кролика, одна из плит подсвечивается красным цветом, и задача игрока сделать так, чтобы кролик оказался на этой плите. Если кролик выбежал за пределы игрового поля, не оказавшись на красной плите, игрок провалил уровень.

Игрок может нажимать на кнопку на игровом автомате. Когда кнопка нажата, все люки на игровом поле открываются, и если кролик находился на люке, он начинает проваливаться вниз. Когда игрок отпускает кнопку, люки закрываются, и кролик продолжает бежать направо. Кролик может провалиться сквозь несколько люков подряд. Если два люка находятся друг под другом, то игрок может успеть нажать и отпустить кнопку так, чтобы кролик провалился в первый люк, но приземлился на уже закрытый люк под ним. Если кролик проваливается с последнего,  $n$ -го этажа, то игрок не прошел уровень. Если кролик оказался на красной плите, то игрок успешно прошел уровень.

Проблема, из-за которой этот автомат не поступил в серийное производство, заключалась в том, что иногда уровень было невозможно пройти. В 1974 году определить, возможно ли пройти уровень, было очень сложной задачей, которую сейчас, в 2020 году, многие школьники способны решить. А сможете ли вы?

### Формат входных данных

В первой строке даны три числа  $n, m$  и  $k$  ( $1 \leq n, m \leq 10^9$ ,  $1 \leq k \leq 2 \cdot 10^5$ ) — число этажей, число плит на каждом этаже, и число люков на игровом поле. Далее  $k$  строк описывают расположение люков. Люк задается координатами  $x_i$  и  $y_i$ , который обозначают, что плита с номером  $y_i$  на этаже  $x_i$  является люком. Никакие два люка не находятся на одной плите.

Далее идет число  $t$  ( $1 \leq t \leq 2 \cdot 10^5$ ) — количество уровней, проходимость которых вам нужно проверить. Каждый уровень описывается двумя парами чисел — стартовыми координатами кролика  $x_j, y_j$  и координатами финишной, красной плиты  $x'_j, y'_j$ . Финишная плита может быть люком.

Координаты люков, стартовых позиций кролика и красных плит являются натуральными числами, и не превосходят  $n$  и  $m$  соответственно.

### Формат выходных данных

Для каждого уровня выведите «Yes» на отдельной строке, если уровень пройти возможно, и «No», если нет.

### Система оценки

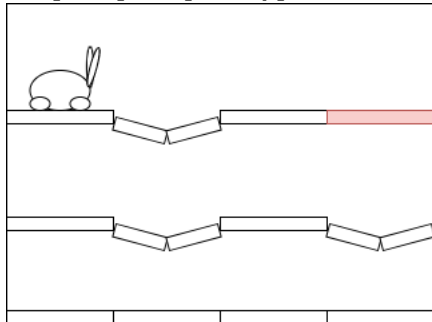
Подзадача	Баллы	Ограничения	Необходимые подзадачи	Результаты
1	23	$n, m, t \leq 200$		Потестовые
2	20	$n, m \leq 2500, t \leq 4000$	1	Потестовые
3	11	$x'_j - x_j = 1$		Потестовые
4	19	$k, t \leq 5000$		Первая ошибка
5	27		1-4	Первая ошибка

### Пример

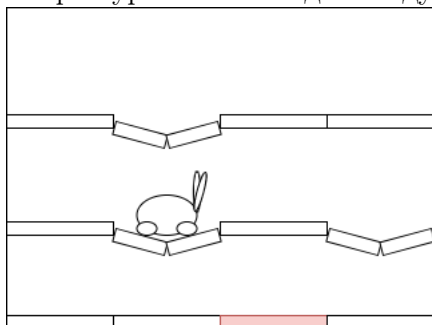
стандартный ввод	стандартный вывод
3 4 3	Yes
1 2	Yes
2 2	No
2 4	
3	
1 1 1 4	
2 2 3 3	
3 2 2 4	

### Замечание

В примере первый уровень выглядит следующим образом:



Если не нажимать кнопку, то кролик будет бежать направо, пока не достигнет красной плитки. Второй уровень выглядит следующим образом:



Если нажать на кнопку сразу, как только появится кролик, он провалится в люк до третьего этажа, и будет бежать направо, пока не достигнет красной плитки.

На третьем уровне кролик никак не сможет подняться на этаж выше.

## Задача 7. Ижевск

Имя входного файла:	<code>input</code>
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Мальчик Миша совсем утомился на карантине. Ему настолько скучно, что сборка и разборка автомата его совсем не интересуют (он уже справляется с ними быстрее всех). Но вот в школе началась новая тема — таблица умножения. В качестве дополнительного домашнего задания «на подумать» учитель задал детям математический кроссворд.

Кроссворд представляет собой прямоугольную таблицу из  $n$  строк и  $m$  столбцов, некоторые клетки которой пустые (их требуется заполнить разгадывающему), а остальные заняты. Строки нумеруются сверху вниз, начиная с единицы, столбцы нумеруются слева направо, начиная с единицы. В отличие от обычного кроссворда, в математическом загадывается не слово, а произведение чисел в некоторой горизонтальной или вертикальной области, то есть перед разгадывающим стоит цель заполнить пустые поля числами так, чтобы в некоторых областях произведение чисел равнялось заданному.

Поскольку решить кроссворд не представляет собой ничего сложного, преподаватель сказал, что пятёрку получит тот, кто сможет при корректной расстановке использовать наименьшее число единиц.

Помогите Мише справиться с этим непростым заданием и получить пятёрку.

### Формат входных данных

Это задача с открытыми тестами, архив тестов вы можете найти на вкладке «Файлы» тестирующей системы.

Входные данные находятся в файлах `01`, `02`, ..., `11`. Каждый файл соответствует описанию одного кроссворда.

Первая строка входного файла содержит имя соответствующего ему выходного файла. Вторая строка содержит три целых числа  $n$ ,  $m$ ,  $k$ , разделённых одиночными пробелами.

Следующие  $n$  строк входного файла описывают таблицу кроссворда: каждая строка состоит ровно из  $m$  символов, где  $j$ -й символ  $i$ -й строки описывает одну клетку, находящуюся в строке  $i$  и столбце  $j$ . Если символ равен «.», то туда Миша должен поставить число, если символ равен «#» — клетка не принадлежит ни одному загаданному произведению.

Следующие  $k$  строк описывают загаданные произведения. Каждое произведение описывается пятью целыми числами  $t$ ,  $x$ ,  $y$ ,  $l$  и  $m$ , разделёнными одиночными пробелами.  $t$  равно 1, если область загаданного произведения — горизонтальная, и равно 2, если вертикальная;  $x$  и  $y$  — номера строки и столбца левого верхнего угла области;  $l$  — длина области в клетках;  $m$  — загаданное произведение. Области одинаковой направленности не могут иметь общих клеток, то есть пересекаться могут только горизонтальная и вертикальная области.

Гарантируется, что существует хотя бы одно решение математического кроссворда из входного файла.

### Формат выходных данных

На проверку необходимо сдать zip-архив, содержащий выходные файлы с названием `01.a`, `02.a`, ..., `11.a`, где выходной файл `XX.a` должен соответствовать входному файлу `XX`. При сдаче в списке языков нужно найти язык «Zip». Выходной файл должен содержать ровно  $n$  строк, в каждой ровно по  $m$  целых неотрицательных чисел, разделённых одиночными пробелами.  $j$ -е число в  $i$ -й строке должно быть равно 0, если  $j$ -й символ в  $i$ -й строке кроссворда равен «#», иначе числу, которое вы ставите в свободную клетку.

Если вы отправляете программу, которая решает задачу, она должна читать из файла с названием `input`, а выводить в стандартный поток. Также, ответы можно отправлять по одному: для этого нужно выбирать текстовый файл в списке языков. Этот текстовый файл подставится в качестве ответа ко всем тестам.

## Система оценки

Если выходной файл не соответствует указанному формату выходных данных, то вы получите 0 баллов за тест. Если построенное решение не удовлетворяет условию задачи, то вы получите 0 баллов за тест. Иначе ваш балл будет вычисляться по формуле  $5 + \lfloor 5 \cdot (\frac{S}{Ans})^2 \rfloor$ , где  $S$  — количество клеток кроссворда, содержащих число большее «1» в вашем решении, а  $Ans$  — количество клеток кроссворда, содержащих число большее «1» в решении жюри.

Результатом по этой задаче является сумма максимальных набранных баллов за каждый тест. Первый тест соответствует тесту из условия, за него баллы не ставятся. Остальные тесты оцениваются из 10 баллов.

В тестирующей системе вам будет доступен результат вашей отправки по каждому тесту.

## Пример

input	стандартный вывод
01. a	2 2 2
3 3 4	3 0 5
...	4 4 4
.#.	
...	
1 1 1 3 8	
1 3 1 3 64	
2 1 1 3 24	
2 1 3 3 40	

## Замечание

Если вы посылаете архив, то обратите внимание, что в архив должен содержать только ответы на тесты, а не папку с ними. Вердикт «Ошибка выполнения» может быть получен, если системе не удастся найти файл с ответом на тест.

## Задача 8. Москва

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

Москва. 20 $xx$  год. Вы директор большой строительной компании. Из мэрии вам пришел новый заказ на реструктуризацию схемы метро. Вы поручили проектирование новой схемы своим лучшим инженерам. И вот, спустя месяц, перед вами лежат  $T$  готовых проектов новой схемы метро. Вы, как большой начальник, должны качественно проанализировать проделанную работу и выбрать лучший проект. Но какой критерий оценивания? В первую очередь, метро – быстрый городской способ передвижения. Поэтому в этой задаче мы предлагаем вам научиться находить расчетное время передвижения между станциями.

Все проекты описаны в одном и том же формате. Есть некоторые станции метро, а также маршруты поездов, каждый из которых состоит ровно из двух различных станций. Во всех маршрутах поезда ходят в обе стороны, то есть отдельно существуют поезда, отправляющиеся от первой станции ко второй, и наоборот. Важной особенностью всех схем является то, что по каждому маршруту поезда отходят от своих начальных станций ровно один раз в  $k$  минут. Да, в 20 $xx$  году метро работает круглосуточно!

Как Вы уже поняли, поезда отправляются от своих начальных станций раз в  $k$  минут, но это, конечно, не означает, что они делают это одновременно. Зафиксируем некоторый начальный момент времени. Тогда для каждого маршрута известно через сколько минут в первый раз по нему отправятся поезда.

Более формально, будем считать, что всего есть  $n$  станций метро, пронумерованных от 1 до  $n$ . Известно  $m$  маршрутов между станциями. Каждый маршрут описывается четверкой чисел:  $u$  – номер начальной станции,  $v$  – номер конечной станции,  $d$  – продолжительность поездки и  $s$  – количество минут, через которое поезда впервые выйдут на этот маршрут. Рассмотрим пример, пусть у нас есть 3 станции, 2 маршрута и  $k = 2$ . Первый из них описывается четверкой  $(1, 2, 3, 1)$ , а второй –  $(2, 3, 1, 0)$ . Тогда для первого маршрута есть два поезда, первый отходит от 1 станции к 2 станции через 1, 3, 5, 7, ... минут, считая от начального момента времени, а продолжительность пути составляет 3 минуты, второй же поезд отправляется от 2 станции к 1 в такие же моменты времени и занимает у него это также 3 минуты.

Конечно, не для каждой пары станций есть маршрут между ними. Поэтому, чтобы добраться от одной станции к другой, необходимо делать пересадки. Пересадки можно делать только на станциях метро. Все поезда ходят строго по расписанию, что позволяет человеку, приехавшему на станцию в некоторый момент времени, успеть на поезд, отходящий от этой же станции в тот же самый момент времени. Вернемся к примеру. Человек, пришедший на 1 станцию в начальный момент времени, может добраться до 3 станции за 5 минут. Для этого необходимо сесть в момент времени 1 на поезд первого маршрута, тогда в момент времени 4, оказавшись на станции 2, можно сесть на поезд второго маршрута и за 1 минуту добраться до станции с номером 3.

Безусловно, вы обладаете большой математической эрудицией, поэтому Вы быстро сообразили, что каждая схема метро представляет собой *кактус*! Кактусом называется **связный** граф, в котором каждая вершина принадлежит не более чем одному простому циклу.

Теперь вы обладаете достаточным набором информации о структуре новых схем метро, чтобы приступить к формальной оценке их качества. Для каждого проекта вы хотите рассчитать минимальное время пути между  $q$  парами станций  $(a, b)$ , при учете, что человек приходит на станцию  $a$  в начальный момент времени.

### Формат входных данных

Первая строка содержит количество проектов  $T$ , переданных вам на оценивание. Далее следует их описание.

Первая строка каждого проекта содержит целые числа  $n$ ,  $m$  и  $k$  ( $1 \leq n \leq 10^5$ ,  $1 \leq m \leq 3 \cdot 10^5$ ,  $1 \leq k \leq 10^9$ ) – количество станций метро, количество маршрутов и интервал движения всех поездов.

Следующие  $m$  строк содержат описания маршрутов — четыре целых числа  $u, v, d$  и  $s$  ( $1 \leq u, v \leq n, u \neq v, 1 \leq d \leq 10^9, 0 \leq s < k$ ) — номера стартовой и конечной станций, продолжительность поездки и время до отправки первого поезда по данному маршруту, считая от начального момента времени. Гарантируется, что между любой парой станций существует не более одного маршрута.

Следующая строка содержит одно целое число  $q$  ( $1 \leq q \leq 10^5$ ) — количество пар станций, между которыми необходимо найти минимальную продолжительность пути. Следующие  $q$  строк содержат по два целых числа  $x$  и  $y$  ( $1 \leq x, y \leq n$ ) — числа, необходимые для генерации номеров станций. В очередной паре первая станция имеет номер  $(x + ans - 1) \bmod n + 1$ , а вторая —  $(y + ans - 1) \bmod n + 1$ , где  $ans$  — ответ для предыдущей пары этого проекта, если она существует, или же 0 в обратном случае.

Гарантируется, что суммарное количество станций во всех проектах не превышает  $10^5$ , а суммарное количество маршрутов —  $3 \cdot 10^5$ . Суммарное количество пар станций, для которых необходимо найти минимальное время пути, во всех проектах также не более  $10^5$ .

### Формат выходных данных

Для каждой пары станций  $(a, b)$  каждого проекта выведите одно целое число — минимальное количество минут, которое необходимо потратить на поездку от станции  $a$  до станции  $b$ .

### Система оценки

Обозначим сумму  $n, m, q$  по всем проектам за  $N, M, Q$  соответственно. Ограничения на  $n, m, k, q$  касаются каждого проекта в отдельности.

Подзадача	Баллы	Ограничения	Необходимые подзадачи	Результаты
1	5	$n = m + 1; 1 \leq N, Q, k \leq 3000$		Потестовые
2	6	$n = m + 1; 1 \leq N, k \leq 3000$	1	Потестовые
3	15	$n = m + 1$	1, 2	Первая ошибка
4	6	$n = m; 1 \leq N, Q, k \leq 3000$		Потестовые
5	10	$n = m$	4	Первая ошибка
6	6	$1 \leq N, k \leq 300$		Потестовые
7	6	$1 \leq N \leq 1000$	6	Потестовые
8	10	$1 \leq N, Q \leq 3000$	1, 2, 4	Первая ошибка
9	36		1–8	Только баллы

### Пример

стандартный ввод	стандартный вывод
<pre> 1 3 2 2 1 2 3 1 2 3 1 0 1 1 3                     </pre>	<pre> 5                     </pre>

### Замечание

Тестовый пример был разобран в условии задачи!