Задача A. Автобусные остановки

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

На улице есть n остановок. Улица представляет собой линию с системой координат. Координаты автобусных остановок: $x_1, x_2, \ldots, x_n$, где x_i — расстояние в метрах от i-й остановки до начала улицы. Первая автобусная остановка находится в начале улицы, а последняя находится в конце улицы.

Существует ровно один автобусный маршрут. Автобусы идут от начала до конца улицы со скоростью v метров в минуту с интервалом в w минут, начиная со времени 0. На каждой остановке останавливается автобус. Остановка не требует времени.

Есть q людей, которые хотят дойти до конца улицы. i-й человек появляется в точке p_i в момент времени t_i и может идти со скоростью не больше u_i метров в минуту. Человек может сесть на автобус и проехать на автобусе. Для каждого человека найдите минимальное время, когда этот человек может добраться до конца улицы.

Формат входных данных

В первой строке дано одно целое число n ($2 \leqslant n \leqslant 150000$) — число остановок.

Во второй строке через пробел дано n целых чисел x_i ($0 = x_1 < x_2 < \ldots < x_n \leqslant 10^9$) — координаты остановок.

В третей строке даны 2 числа w и v ($1 \leqslant w, v \leqslant 10^9$) — интервал и скорость движения автобусов соответственно.

В четвёртой строке дано число q ($1 \leqslant q \leqslant 150000$) — число людей.

В следующих q строках идёт описание людей. В i-й из них записаны 3 числа p_i, t_i, u_i ($0 \leqslant p_i < x_n$, $0 \leqslant t_i \leqslant 10^9$, $1 \leqslant u_i \leqslant 10^9$) — координата точки появления i-го человека, время его появления и его скорость.

Формат выходных данных

Выведите q строк. В i-й строке выведите минимальное время, когда i-й человек сможет добраться до конца улицы.

Ответ считается корректным, если его абсолютная погрешность не превышает 10^{-6}.

Система оценки

Подзадача на 50 баллов:

- $n, q \leqslant 3000$

Пример

стандартный ввод	стандартный вывод
4 0 10 40 100 20 10 3 0 0 4 15 10 1 40 2 16	10 30 5.75

Задача B. Ретро

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

Маленький Мирко получил игровую консоль на Рождество. Это не Playstation 4 и не Xbox one, а Atari 2600, на которой находилась одна бесплатная игра. Протагонист игры стоял внизу экрана, сверху появлялись различные объекты, которые падали вниз.

Говоря точнее, экран представлен в виде поля $R \times S$ пикселей, R строк и S столбцов. Протагонист занимал один пиксель, помеченный «`M`», и находился на нижней строчке поля. Остальные пиксели были помечены одним из следующих символов: «`.`» (пустая клетка), «`*`» (бомба), «`(`» (открывающая скобка), «`)`» (закрывающая скобка).

Протагонист мог перемещаться влево или вправо на один пиксель, или оставаться на месте, в то время как остальные объекты одновременно перемещаются на один пиксель вниз (возможно за экран). Когда персонаж попадает на скобку, она записывается в специальный массив. В конце игры требуется собрать в этом массиве максимально возможную по длине **правильную** скобочную последовательность.

Правильная скобочная последовательность (далее ПСП) определяется по следующим правилам:

1. «`()`» является ПСП.
2. Если `A` — ПСП, то «`(`A`)`» тоже является ПСП.
3. Если `A` и `B` — ПСП, то «`AB`» тоже является ПСП.

Игра заканчивается, если позиция игрока совпала с позицией бомбы, или когда все объекты упали за экран.

Формат входных данных

В первой строке вводятся натуральные числа R и S ($1 \leqslant R, S \leqslant 300$) — размеры поля. В каждой из следующих R строк вводятся S символов «`M`», «`.`», «`*`», «`(`» или «`)`» — игровое поле.

Формат выходных данных

В первой строке выведите длину максимальной скобочной последовательности, которую Мирко может получить. Во второй строке выведите это последовательность. Если ответов несколько, выведите **лексикографически минимальный** из них.

Система оценки

Программы, верно работающие при $R \leqslant 15$ оцениваются в 25 баллов. Программы, верно работающие при $R \leqslant 100$ оцениваются в 50 баллов.

Примеры

стандартный ввод	стандартный вывод
5 4 ..). .)(. (.)* *(.* ..M.	4 (())
6 3)(. *.. (**)() (). M..	4 ()()
6 3 ((. *.. (**)() (). M..	2 ()

Замечание

Пояснения к первому примеру: движения протагониста такие: влево, влево, вправо, вправо.

Пояснения ко второму примеру: движения протагониста такие: не двигаться, не двигаться, не двигаться, вправо, влево.

Пояснения ко третьему примеру: движения протагониста такие: не двигаться, не двигаться, вправо.

Задача C. Ханы

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Элли недавно узнала про Булгарских ханов — правителей кочевых народов, которые путешествовали по континенту сотни лет, прежде чем они наконец поселились навсегда в местах, где сейчас находится Болгария.

Континент, по которому они скитались, разделен на $N * M$ регионов, расположенных в форме прямоугольника с N строками и M столбцами. Ханы останавливались ровно на один год в определенном регионе, и пока они жили там, их клан съедал всю еду в этом регионе. В конце года они перемещались в один из (не более чем) четырех соседних по стороне регионов, там они проводили следующий год, съедая всю еду в нем, и так далее. Будем считать, что перемещения в соседний регион происходят мгновенно (в конце концов, что такое несколько дней путешествия по сравнению с целым годом). Ханы никогда не проводили два года подряд в одном регионе, в этом случае их клан мог бы погибнуть от голода.

Для каждого региона известно максимальное количество еды, которое может в нем находиться. Обозначим это значение целым числом A_{ij}.После того, как ханы съедали всю еду в регионе и уезжали со своим кланом в соседний регион, еда в нем начинала восстанавливаться. Через год после отъезда ханов в регионе появлялась 1 единица еды. После этого каждый год количество еды в регионе удваивалось, пока оно не достигало максимального значения A_{ij} для этого региона. Обратите внимание, что количество еды никогда не превышало максимального количества, которое могло находиться в регионе. Например, если $A_{ij} = 55$, то количество еды в этом регионе в начале каждого из первых десяти лет после отъезда ханов из этого региона, было, соответственно, $0, 1, 2, 4, 8, 16, 32, 55, 55, 55$.

Ханы никогда не возвращались в регион, пока он не восстанавливался полностью и количество еды в нем не достигало максимума. Из-за этого могла, например, сложиться ситуация, что ханы переместились в регион, где меньше еды (скажем, 42 единицы, но это максимальное количество), а не в регион, где больше еды (скажем, 64, а максимум 71). В примере в предыдущем параграфе ханы могли бы вернуться в регион в начале 8 года после своего отъезда, это первый год, в который в этом регионе количество еды максимально.

Элли знает информацию о максимальном количестве еды в каждом регионе континента, заданную как матрицу A с N строками и M столбцами. Зная, что ханы проведут первый год в левом верхнем регионе, а исходно каждый регион содержит максимальное возможное для этого региона количество еды, выясните, какое максимальное количество еды ханы смогут суммарно съесть за K лет.

Формат входных данных

На первой строке входных данных заданы три целых числа N, M, и K ($1 \leqslant N, M \leqslant 10$, $1 \leqslant K \leqslant 100$), задающих, соответственно, количество строк, количество столбцов в матрице и количество лет. На каждой из следующих N строк находятся по M целых чисел A_{ij} ($10 \leqslant A_{ij} \leqslant 100$), задающих максимальное количество еды в соответствующем регионе.

Формат выходных данных

Выведите одну строку, содержащую одно целое число — максимальное суммарное количество еды, которое ханы смогут съесть, если они будут путешествовать оптимально.

Гарантируется, что всегда будет путь, который не нарушает правила, что нельзя повторно посещать регион до момента, когда в нем полностью восстановится максимальное количество еды.

Система оценки

В данной задаче каждый тест оценивается отдельно

- В тестах, имеющих стоимость в 20 процентов от баллов за задачу, выполнено $1 \leqslant N, M \leqslant 4$

- В тестах, имеющих стоимость в еще 20 процентов от баллов за задачу, выполнено $1 \leqslant K \leqslant 20$

Пример

стандартный ввод	стандартный вывод
4 4 11 11 17 13 96 10 12 18 15 13 12 16 17 24 10 14 22	254

Замечание

В первом примере регионы, которые ханы могут посетить, чтобы съесть максимальное количество еды (254 единицы) — регионы с максимальным количеством еды $11, 17, 13, 96, 15, 17, 22, 14, 16, 18, 15$, соответственно. При этих перемещениях они посетят дважды лишь один регион — с максимальным количеством еды 15. Обратите внимание, что после последнего года все регионы, соседние с последним регионом, посещенном ханами, не содержат максимального возможного количества еды. Для приведенного теста это не проблема, поскольку это последний год. Но если бы ханам необходимо было продолжить перемещения (например, K было бы равно 12, а не 11), то им пришлось бы выбрать другой путь. Вариант оптимального пути для K =12 по континенту из первого примера: $11, 17, 13, 96, 15, 18, 16, 17, 22, 14, 10, 24$, с суммой 273.

Задача D. Ghiță, Lică Sămădăul и Buză Spartă

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

«Ghiţă» очень любит программирование. Его любимые занятия — играть с перестановками и проводить время со своей женой Зинаидой. На свою 10-летнюю годовщину свадьбы Зинаида подарила ему очень красивую перестановку, ведь она знала, что это лучший подарок, который может получить «Ghiţă». Пусть P_j — это j-элемент перестановки для каждого $1 \leqslant j \leqslant N$.

«Ghiţă» был так рад такому подарку, что начал вычислять значение Q_i для каждого i что $1 \leqslant i \leqslant N$. Q_i — это число возрастающих подпоследовательностей на префиксе длины i в перестановке.

Более формально для каждого $1 \leqslant i \leqslant N$, Q_i равняется числу последовательностей $j_1, j_2, \dots j_k$, что $1 \leqslant j_1 < j_2 < \dots < j_{k-1} < j_k \leqslant i$ и $P_{j_1} < P_{j_2} < \dots < P_{j_k}$.

Q конечно не перестановка, но тоже крутая штука, поэтому «Ghiţă» положил её рядом с P.

Все было нормально, пока не появился «Lică Sămădăul». Он хотел использовать систему эпиднадзора «Ghiţă» для аморальных целей, а «Ghiţa», будучи честным человеком, не помог ему. Разгневанный ответом «Ghiţă», «Lică Sămădăul» нанял «Buză Spartă», чтобы помочь ему украсть самые ценные активы «Ghiţă»: его перестановку и его жену. И так он и сделал.

На следующий день «Ghiţă» выяснил, что P отсутствует, и теперь единственная возможность для «Ghiţă» восстановить перестановку — это использовать массив Q, который у него есть. Как вы уже догадались, ваша задача — помочь «Ghiţă» восстановить перестановку P, используя массив Q.

Формат входных данных

В первой строке ввода записано одно число N ($1 \leqslant N \leqslant 70\,000$).

Во второй строке через пробел записаны N целых чисел $Q_1, Q_2, \dots, Q_N$.

Размер входного файла не превышает **115 МБ**.

Мы советуем вам самостоятельно проверять время работы и использование памяти читающей части вашей программы, чтобы убедиться, что возможная неэффективность вашей программы не связана с этой частью.

Формат выходных данных

В первой и единственной строке выведите P — украденную перестановку.

Гарантируется, что существует ровно один возможный ответ (только один P имеет заданный Q).

Система оценки

№	Баллы	Ограничения	T = длине Q_i	Размер входных данных
0	0	Тесты из условия	—	—
1	10	$N \leqslant 9$	—	—
2	15	$N \leqslant 400$	$T \leqslant 18$	—
3	18	$N \leqslant 700$	—	—
4	17	$N \leqslant 40\,000$	$T \leqslant 171$	4.5 МБ
5	11	$N \leqslant 70\,000$	$T \leqslant 258$	10 МБ
6	7	$N \leqslant 70\,000$	$T \leqslant 314$	16 МБ
7	16	$N \leqslant 70\,000$	—	85 МБ
8	6	$N \leqslant 70\,000$	—	115 МБ

Примеры

стандартный ввод	стандартный вывод
4 1 2 5 6	3 2 4 1
6 1 3 5 9 11 21	1 6 3 4 2 5

Замечание

В первом примере $N = 4$ и $P = \{3, 2, 4, 1\}$

$Q_1 = 1$, так как $\{3\}$ — единственная возрастающая подпоследовательность $\{3\}$

$Q_2 = 2$, потому что $\{3\}$ и $\{2\}$ — единственные возрастающие подпоследовательности $\{3, 2\}$

$Q_3 = 5$, потому что $\{3\}, \{3, 4\}, \{2\}, \{2, 4\}\{4\}$ — единственные возрастающие подпоследовательности $\{3, 2, 4\}$

$Q_4 = 6$, потому что $\{3\}, \{3, 4\}, \{2\}, \{2, 4\}, \{4\}\{1\}$ — единственные возрастающие подпоследовательности $\{3, 2, 4, 1\}$.