

Задача А. Наша дружба крепче камня

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

У Вовы есть n камней для игры в ним, пронумерованных от 1 до n . Он уже досконально изучил игру, поэтому придумал себе другое развлечение. У каждого камня есть свой изначальный цвет a_i , но они обладают редкими природными свойствами, поэтому умеют менять свой цвет со временем.

Теперь Вова выполнит n операций, по одной для каждого камня. Он будет выкладывать камни на стол в одну линию слева направо. Операция i ($1 \leq i \leq N$) выполняется так:

1. Вова кладёт i -й камень сразу справа от $i - 1$ -го камня. Если $i = 1$, то он просто кладёт камень 1 на стол.
2. Если среди камней $1, 2, \dots, i - 1$ есть камень, чей текущий цвет совпадает с цветом камня i , то происходит особая реакция и какие-то камни меняют цвет. Пусть j — максимальный индекс такого камня, чей текущий цвет совпадает с цветом камня i . Тогда все камни с индексами $j + 1, \dots, i - 1$ меняют цвет на a_i .

Вова очень нетерпелив и хочет заранее узнать, какими будут цвета всех камней после выполнения всех n операций. Помогите Вове и по заданным начальным цветам определите итоговые цвета всех камней.

Формат входных данных

Первая строка содержит одно целое число n ($1 \leq n \leq 200\,000$) — количество камней.

Следующие n строк содержат по одному целому числу a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — изначальные цвета камней.

Формат выходных данных

Выведите n строк. В i -й строке выведите итоговый цвет камня i после выполнения всех операций.

Система оценки

Подгруппа	Баллы	Дополнительные ограничения	Необ. подгруппы
0	0	Тесты из условия	
1	25	$n \leq 2\,000$	0
1	35	$a_i \leq 2$	
3	40	Без дополнительных ограничений	0–2

Примеры

стандартный ввод	стандартный вывод
6	1
1	1
2	1
1	2
2	2
3	2
2	
10	1
1	1
1	1
2	1
2	1
1	1
2	1
2	1
1	1
1	2
2	

Замечание

Рассмотрим первый пример из условия; операции происходят так:

- После операции 1: на столе лежит [1].
- После операции 2: на столе лежит [1, 2].
- После операции 3: сначала получаем [1, 2, 1]. Последний камень того же цвета слева — это камень 1, поэтому камень 2 перекрашивается в 1. Получаем [1, 1, 1].
- После операции 4: на столе лежит [1, 1, 1, 2].
- После операции 5: на столе лежит [1, 1, 1, 2, 3].
- После операции 6: сначала получаем [1, 1, 1, 2, 3, 2]. Последний камень того же цвета слева — это камень 4, поэтому камень 5 перекрашивается в 2. Получаем [1, 1, 1, 2, 2, 2].

Задача В. Самый маленький элемент

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

Это интерактивная задача.

У Андрея изначально есть пустой массив a , затем происходит n запросов добавления элементов к нему. В i -м запросе в конец массива добавляют ровно x_i новых элементов, причем их значения не известны. После этого Андрей хочет найти минимальный элемент среди всех **ещё не удалённых** элементов массива, а затем сразу удалить его. При удалении индексы остальных элементов не меняются.

Проблема в том, что сами значения элементов вам неизвестны, поэтому можно задавать вопросы сравнения двух элементов.

Андрей не может найти самый маленький элемент, поскольку не знает сами значения массива a , поэтому ваша задача — после каждого запроса добавления новых элементов определить индекс текущего минимального элемента.

Протокол взаимодействия

Первая строка содержит одно целое число n ($1 \leq n \leq 40$) — количество запросов.

Далее следуют данные интерактора для каждого запроса.

Сначала для очередного запроса подаётся одно целое число x_i ($1 \leq x_i \leq 2000$) — число элементов, которые нужно добавить в конец массива. Гарантируется, что суммарная длина массива за всё взаимодействие не превосходит 2000.

После этого вы можете сравнивать элементы с помощью запросов вида $? i j$, и интерактор выводит:

- 0, если $a_i < a_j$;
- 1, если $a_i > a_j$;

При этом должно выполняться $i \neq j$ и индексы i и j не должны соответствовать уже удалённым элементам. Гарантируется, что все элементы массива попарно **различны**.

Когда минимальный элемент найден, выводите: $! x$, где x — индекс минимального среди ещё не удалённых элементов. После этого считается, что элемент с индексом x удалён, и начинается следующий запрос.

После каждого вывода не забывайте делать сброс буфера. После обработки последнего запроса взаимодействие заканчивается. Итоговый балл зависит от общего числа заданных вопросов.

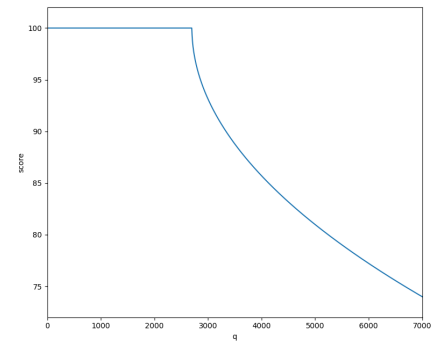
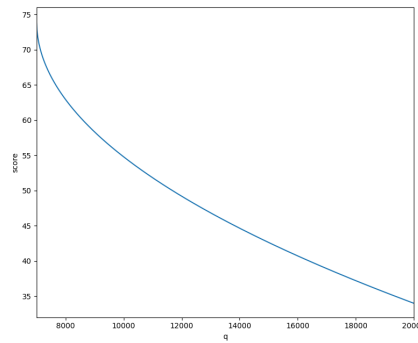
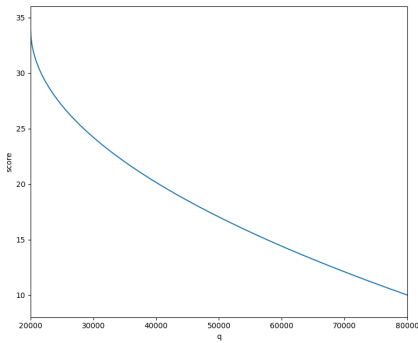
Если ваше решение не будет соответствовать формату вывода, или вы сделаете слишком много запросов, то вы можете получить любой некорректный вердикт.

Система оценки

Пусть q — общее число вопросов вида $? i j$, которые задала ваша программа.

Тогда количество баллов вычисляется по формуле

$$\text{score}(q) = \begin{cases} 100, & q \leq 2700, \\ 100 - 26\sqrt{\frac{q - 2700}{7000 - 2700}}, & 2700 < q \leq 7000, \\ 74 - 40\sqrt{\frac{q - 7000}{2 \cdot 10^4 - 7000}}, & 7000 < q \leq 2 \cdot 10^4, \\ 34 - 24\sqrt{\frac{q - 2 \cdot 10^4}{8 \cdot 10^4 - 2 \cdot 10^4}}, & 2 \cdot 10^4 < q \leq 8 \cdot 10^4, \\ 10, & q > 8 \cdot 10^4. \end{cases}$$



На графиках показан балл за задачу в зависимости от количества запросов. Например, для $q = 9207$ решение будет набирать 57 баллов.

Пример

стандартный ввод	стандартный вывод
3	? 1 2
3	? 1 3
	? 2 3
1	! 2
0	? 1 4
0	! 4
	? 1 5
1	! 1
1	
1	
0	

Замечание

В первом примере загаданный массив $[3, 2, 4, 1, 5]$. Сначала добавляется 3 элемента. Первым запросом узнаем, что $a_1 > a_2$, вторым $a_1 < a_3$, третьим $a_2 < a_3$, поэтому минимальный элемент a_2 .

Задача С. Леонардо, Донателло, Рафаэль и Микеланджело

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

Леонардо устроил черепашью мозгом внеплановую тренировку. Перед командой лежит прямоугольная карта канализации размером $n \times m$. В клетке с координатами (x, y) записано число $a_{x,y}$ — это стоимость прохода через этот участок. Где-то грязь, где-то люк, где-то подозрительно скользко после очередной драки с кланом Фут, а в каких-то клетках вообще пицца. Поэтому значения $a_{x,y}$ бывают как положительные, так и отрицательные.

Микеланджело выбрал q маршрутов и хочет знать, сколько минимум усилий нужно потратить на каждый из них. В каждом маршруте заданы две клетки: (x_s, y_s) и (x_t, y_t) , причём гарантируется, что $x_s \leq x_t$ и $y_s \leq y_t$.

Черепашки хоть и передвигаются по-боевому, но вполне предсказуемо: из клетки можно пойти только вниз или вправо. Для каждого запроса требуется найти минимальную возможную сумму чисел в клетках на пути из (x_s, y_s) в (x_t, y_t) .

Формально, нужно минимизировать сумму $a(x_1, y_1) + a(x_2, y_2) + \dots + a(x_k, y_k)$ по всем последовательностям клеток $\langle (x_1, y_1), \dots, (x_k, y_k) \rangle$, для которых выполняется:

- $x_1 = x_s$ и $y_1 = y_s$;
- $x_k = x_t$ и $y_k = y_t$;
- для каждого $1 \leq i < k$ следующая клетка получается либо переходом вниз, либо переходом вправо, то есть $(x_{i+1}, y_{i+1}) = (x_i + 1, y_i)$ или $(x_{i+1}, y_{i+1}) = (x_i, y_i + 1)$.

Помогите черепашкам быстро отвечать на такие запросы, чтобы у Донателло осталось время на технику, у Рафаэля — на ворчание, а у Микеланджело — на пищу.

Формат входных данных

Первая строка содержит три целых положительных числа n , m и q ($1 \leq n \cdot m, q \leq 200\,000$) — количество строк, количество столбцов и число маршрутов.

В следующих n строках записана таблица, в i -й из них содержатся m целых чисел $a_{i,1}, a_{i,2}, \dots, a_{i,m}$ ($|a_{i,j}| \leq 10^4$) — стоимость прохода через клетку.

Далее идут q строк с описанием маршрутов. Каждая строка содержит четыре целых числа x_s, y_s, x_t, y_t ($1 \leq x_s \leq x_t \leq n, 1 \leq y_s \leq y_t \leq m$) — начальная и конечная клетки очередного маршрута.

Формат выходных данных

Для каждого маршрута выведите в отдельной строке одно число — минимальную сумму на пути из (x_s, y_s) в (x_t, y_t) .

Система оценки

Подгруппа	Баллы	Доп. ограничения	Необх. группы
0	0	Тесты из условия	
1	6	$n \cdot m \leq 2500, q \leq 1000$	0
2	11	$n \cdot m \leq 10\,000$	0, 1
3	5	$n \cdot m \leq 200\,000, n = 1$	—
4	18	$n \cdot m \leq 200\,000, n = 5$	—
5	20	$n \cdot m \leq 50\,000$	0, 1, 2
6	27	$n \cdot m \leq 100\,000$	0, 1, 2, 5
7	13	—	0–6

Пример

стандартный ввод	стандартный вывод
5 5 6	7
1 2 1 1 1	7
1 1 3 2 2	5
1 1 1 1 1	7
1 3 3 1 1	6
2 3 3 1 1	9
1 1 4 4	
2 2 5 5	
1 1 1 4	
1 1 2 4	
2 2 4 3	
1 1 5 5	

Задача D. Беспилотный транспорт

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

В Берляндии в сутках ровно S минут, а момент через x минут после начала суток называется временем x , где $0 \leq x < S$. В Берляндии есть N городов, пронумерованных от 0 до $N - 1$, и M дорог, пронумерованных от 0 до $M - 1$. По дорогам можно добраться из любого города в любой другой. Дорога i соединяет города A_i и B_i , и проезд по ней в любом направлении занимает L_i минут.

В Яндексе тестируют сеть беспилотного транспорта между городами Берляндии. Города соединены дорогами, по которым автономные машины могут ездить в обе стороны. Но не всё так просто: на некоторых дорогах каждый день в определённый момент начинается техническое окно — выезд беспилотников на такую дорогу после этого момента запрещён до конца суток. То есть для i -й дороги с момента времени C_i и до конца суток проехать никак нельзя. Беспилотник может начать движение по дороге i в момент x только если он успеет полностью завершить проезд до начала технического окна, то есть если выполнено $x + L_i \leq C_i$.

Если машина не успевает — придётся подождать в городе до следующего подходящего момента. Стоять в городах можно сколько угодно: зарядка есть, кофе для инженеров тоже, паниковать не нужно.

Вам поручили помочь команде беспилотного транспорта в базовой задаче расчета времени поездки. Есть Q запросов. Для каждого запроса j известно, из какого города U_j и в какой момент времени T_j стартует машина, а также в какой город V_j ей нужно попасть. Требуется определить, за какое минимальное время машина сможет добраться до цели.

Формат входных данных

В первой строке входных данных содержится четыре целых числа N, M, S, Q ($1 \leq N \leq 90, N - 1 \leq M \leq \frac{N(N-1)}{2}, 2 \leq S \leq 10^{15}, 1 \leq Q \leq 10^6$) — количество городов в Берляндии, дорог между этими городами, количество минут в сутках и количество запросов.

В следующих M строках входных данных содержится описание дорог A_i, B_i, L_i, C_i ($0 \leq A_i, B_i < N, 1 \leq L_i < C, L_i \leq C_i < S$) — два города, которые соединены дорогой, длительность проезда по дороге и время перекрытия дороги для беспилотников. Гарантируется, что каждая дорога встречается не более одного раза в вводе.

В следующих Q строках входных данных содержится описание запросов U_j, V_j, T_j ($0 \leq U_j, V_j < n, 0 \leq T_j < S, U_j \neq V_j$) — начальная и финальная точки поездки, и время старта поездки.

Формат выходных данных

Выведите Q строк. В j -й строке должно находиться одно целое число — минимальное время, необходимое участнику, чтобы добраться из города U_j в город V_j , начиная в момент времени T_j . Заметьте, что вам нужно посчитать именно время поездки.

Система оценки

Тесты разбиты на 5 подгрупп. Баллы за подгруппу начисляются только если пройдены все тесты этой подгруппы и всех необходимых предыдущих подгрупп.

Подгруппа	Баллы	Дополнительные ограничения	Необходимые подгруппы
0	0	Тесты из условия	—
1	15	$N \leq 40, Q \leq 1000$	0
2	20	$N \leq 40, U_j = 0$ для всех j	—
3	20	$N \leq 40$	0–2
4	30	$N \leq 60$	0–3
5	15	Нет дополнительных ограничений	0–4

Примеры

стандартный ввод	стандартный вывод
4 5 20 6	3
0 1 3 19	8
0 2 2 8	14
1 2 4 15	2
1 3 5 14	5
2 3 1 18	7
0 3 5	
0 3 7	
0 3 9	
2 0 6	
3 1 10	
1 2 15	
6 10 100 9	42
5 3 4 29	32
1 0 6 26	4
0 4 2 7	93
0 5 18 18	99
2 0 79 82	6
3 4 35 46	102
1 2 15 57	60
2 4 3 6	39
4 1 21 83	
3 2 47 53	
0 2 63	
0 4 70	
0 4 98	
0 5 25	
0 5 19	
0 4 96	
0 5 2	
0 3 62	
0 3 83	

Замечание

Участник может ждать в городах сколько угодно. Если в текущие сутки на нужную дорогу уже поздно выезжать, можно спокойно переночевать в городе и поехать на следующий день. Да, тайные операции тоже иногда требуют здорового сна и нормального расписания.