

Задача А. Задача Иосифа

Имя входного файла: joseph.in
Имя выходного файла: joseph.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

N мальчиков стоят по кругу. Они начинают считать себя по часовой стрелке, счет ведется с единицы. Как только количество посчитанных достигает p , последний посчитанный (p -й) мальчик покидает круг, а процесс счета начинается со следующего за ним мальчика и вновь ведется с единицы.

Последний оставшийся в кругу выигрывает.

Можете ли вы посчитать, номер выигравшего мальчика в исходном кругу? (мальчики нумеруются числами от 1 до N по часовой стрелке, начиная с того самого мальчика, с которого начинался счет).

Формат входных данных

Во входном файле два целых числа — N и P ($1 \leq N, P \leq 10^5$).

Формат выходных данных

Выведите номер выигравшего мальчика.

Примеры

joseph.in	joseph.out
3 4	2

Задача В. Учиться!

Имя входного файла: `moscow.in`
Имя выходного файла: `moscow.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Каждый год огромное количество выпускников, сдавшие ЕГЭ, выбирают, куда же они пойдут учиться. Не удивительно, что многие из них предпочитают перебраться поближе к столице. Транспортная инфраструктура страны переживает не лучшие времена, и в приемлемом качестве поддерживается минимально возможное число городов, необходимое для того, чтобы от любого города можно было добраться до любого другого.

Каждый выпускник оценивает свои результаты сдачи экзаменов, и решает, насколько далеко от своего родного города в сторону столицы он сможет уехать.

Выпускников настолько много, что вам не требуется выводить для каждого из них, до какого города он сможет доехать. Достаточно вывести сумму ответов для каждого выпускника.

Запросы генерируются следующим образом. Заданы числа a_1, a_2 и числа x, y и z . Числа a_3, \dots, a_{2m} генерируются следующим образом: $a_i = (x \cdot a_{i-2} + y \cdot a_{i-1} + z) \bmod n$. Первый запрос имеет вид $\langle a_1, a_2 \rangle$. Если ответ на $i - 1$ -й запрос равен v , то i -й запрос имеет вид $\langle (a_{2i-1} + v) \bmod n, a_{2i} \rangle$. В i -м запросе первое число соответствует городу, в котором окончил школу i -й выпускник, а второе — насколько далеко от родного города он может уехать. Все выпускники стараются перебраться как можно ближе к столице.

Формат входных данных

Первая строка содержит два числа: n ($1 \leq n \leq 100\,000$) и m ($1 \leq m \leq 10\,000\,000$). Столица имеет номер 0. Вторая строка содержит $n - 1$ целых чисел, i -е из этих чисел равно номеру следующего за городом i на пути к столице. Третья строка содержит два целых числа в диапазоне от 0 до $n - 1$: a_1 и a_2 . Четвертая строка содержит три целых числа: x, y и z , эти числа неотрицательны и не превосходят 10^9 .

Формат выходных данных

Выведите в выходной файл сумму номеров городов — ответов на все запросы.

Примеры

<code>moscow.in</code>	<code>moscow.out</code>
3 2 0 1 2 1 1 1 0	1
1 2 0 0 1 1 1	0

Задача С. Аполлинару

Имя входного файла: `apollinaria.in`
Имя выходного файла: `apollinaria.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Паша завтра читает лекцию по базовым алгоритмам на строках. В частности, ему придется рассказать ЛКШатам про Z -функцию. Сам он когда-то знал, что это такое, но с тех пор успел многое забыть.

Готовясь к лекции, он открыл <http://ru.wikipedia.org/wiki/Z-функция> и даже http://e-maxx.ru/alg/z_function. Теперь Паша вспомнил, что Z -функция от строки S — это массив z , каждый элемент которого z_i равен длине самого длинного префикса подстроки, начинающейся с позиции i в строке S .

Так, например, для строки `aabcaabxaaaaz` Z -функция это $[12, 1, 0, 0, 3, 1, 0, 0, 2, 2, 1, 0]$. Ее суммарное значение равно 22.

В группе у Паши учится девочка Аполлинару, которая любит задавать каверзные вопросы и, чтобы не попасть впросак, Паша прорешивает различные задачи по теме лекции. Сейчас он задумался над очередной. В заданной строке требуется переставить буквы и сделать так, чтобы суммарное значение Z -функции было минимальным.

Формат входных данных

Во входном файле содержится непустая строка, состоящая из строчных латинских букв. Гарантируется, что каждая буква встречается не более 1000 раз.

Формат выходных данных

В выходной файл выведите строку состоящую из тех же символов, что и входная строка. Причем суммарное значение Z -функции для выходной строки должно быть минимальным. Если ответов несколько, выведите **минимальную** лексикографически строку.

Примеры

<code>apollinaria.in</code>	<code>apollinaria.out</code>
<code>apollinaria</code>	<code>naaaillopr</code>

Задача D. Уничтожение массива

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вам дан массив, состоящий из n неотрицательных целых чисел a_1, a_2, \dots, a_n .

В этом массиве один за другим зачёркиваются числа. Вам задана перестановка чисел от 1 до n — порядок, в котором это происходит.

После зачёркивания очередного числа вам необходимо найти в этом массиве подотрезок с максимальной суммой, не содержащий ни одного зачёркнутого числа. Сумму чисел в пустом подотрезке считайте равной 0.

Формат входных данных

В первой строке входных данных записано число n ($1 \leq n \leq 100\,000$) — длина массива.

В второй строке записаны n целых чисел a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^9$).

В третьей строке входных данных записана перестановка чисел от 1 до n — порядок, в котором зачеркиваются числа.

Формат выходных данных

В выходной файл выведите n строк, каждая из которых должна содержать одно число — максимальную сумму на подотрезке заданного массива, не содержащем зачёркнутых чисел, после выполнения очередного действия.

Примеры

стандартный ввод	стандартный вывод
4 1 3 2 5 3 4 1 2	5 4 3 0
5 1 2 3 4 5 4 2 3 5 1	6 5 5 1 0
8 5 5 4 4 6 6 5 5 5 2 8 7 1 3 4 6	18 16 11 8 8 6 6 0

Замечание

В первом тестовом примере происходит следующее:

- Зачеркивается третий элемент, массив принимает вид 1 3 * 5. Отрезок с максимальной суммой 5 состоит из одного числа 5.
- Зачеркивается четвертый элемент, массив принимает вид 1 3 * *. Отрезок с максимальной суммой 4 состоит из двух чисел 1 3.
- Зачеркивается первый элемент, массив принимает вид * 3 * *. Отрезок с максимальной суммой 3 состоит из одного числа 3.

4. Зачеркивается оставшийся второй элемент, в этот момент непустых допустимых подотрезков не остается, поэтому здесь ответ равен нулю.

Задача Е. Дубы

Имя входного файла:	<code>oaks.in</code>
Имя выходного файла:	<code>oaks.out</code>
Ограничение по времени:	1.5 секунды
Ограничение по памяти:	256 мегабайт

На аллее перед зданием Министерства Обороны в ряд высажены n дубов. В связи с грядущим приездом главнокомандующего, было принято решение срубить несколько деревьев для придания аллее более милитаристического вида.

Внутренние распоряжки министерства позволяют срубить дуб только в двух случаях:

- если и ближайший дуб слева, и ближайший дуб справа строго ниже, чем данный дуб;
- если и ближайший дуб слева, и ближайший дуб справа строго выше, чем данный дуб.

В частности, согласно этому правилу, нельзя срубить крайний левый и крайний правый дубы.

Министр хочет выработать такой план вырубki, чтобы в итоге осталось несколько дубов, высоты которых образуют неубывающую последовательность, то есть чтобы каждый дуб был не ниже, чем все дубы, стоящие слева от него. При этом, как человек любящий флору, министр хочет, чтобы было срублено минимальное возможное количество деревьев.

Помогите сотрудникам министерства составить оптимальный план вырубki аллеи или выяснить, что срубить дубы соответствующим образом невозможно.

Формат входных данных

Первая строка входного файла содержит целое число n — количество дубов, растущих на аллее ($2 \leq n \leq 200$). Вторая строка содержит n чисел — высоты дубов, приведённые слева направо. Высоты дубов — положительные целые числа, не превышающие 1 000.

Формат выходных данных

Если оставить последовательность дубов с неубывающими высотами невозможно, выходной файл должен содержать только одно число -1 .

В случае, если искомый план существует, в первую строку выходного файла выведите целое число m — минимальное количество дубов, которые необходимо срубить. В следующие m строк выведите оптимальный план вырубki деревьев — номера дубов в том порядке, в котором их следует срубить, по одному номеру на строке.

Дубы нумеруются слева направо натуральными числами от 1 до n .

Если планов с наименьшим числом срубаемых дубов несколько, выведите любой из них.

Примеры

<code>oaks.in</code>	<code>oaks.out</code>
5	2
3 2 4 8 5	2
	4

Задача F. Эх, дороги . . .

Имя входного файла:	roads.in
Имя выходного файла:	roads.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В многострадальном Тридесятом государстве опять готовится дорожная реформа. Впрочем, надо признать, дороги в этом государстве находятся в довольно плачевном состоянии. Так что реформа не повредит. Одна проблема — дорожникам не развернуться, поскольку в стране действует жесткий закон — из каждого города должно вести не более двух дорог. Все дороги в государстве двусторонние, то есть по ним разрешено движение в обоих направлениях (разумеется, разметка отсутствует). В результате реформы некоторые дороги будут строиться, а некоторые другие закрываться на бессрочный ремонт.

Петя работает диспетчером в службе грузоперевозок на дальние расстояния. В связи с предстоящими реформами, ему необходимо оперативно определять оптимальные маршруты между городами в условиях постоянно меняющейся дорожной ситуации. В силу большого количества пробок и сотрудников дорожной полиции в городах, критерием оптимальности маршрута считается количество промежуточных городов, которые необходимо проехать.

Помогите Пете по заданной последовательности сообщений об изменении структуры дорог и запросам об оптимальном способе проезда из одного города в другой, оперативно отвечать на запросы.

Формат входных данных

В первой строке входного файла заданы числа n — количество городов, m — количество дорог в начале реформы и q — количество сообщений об изменении дорожной структуры и запросов ($1 \leq n \leq 100\,000$, $0 \leq m \leq 100\,000$, $0 \leq q \leq 200\,000$). Следующие m строк содержат по два целых числа каждая — пары городов, соединенных дорогами перед реформой. Следующие q строк содержат по три элемента, разделенных пробелами. «+ $i j$ » означает строительство дороги от города i до города j , «- $i j$ » означает закрытие дороги от города i до города j , «? $i j$ » означает запрос об оптимальном пути между городами i и j .

Формат выходных данных

На каждый запрос вида «? $i j$ » выведите одно число — минимальное количество промежуточных городов на маршруте из города i в город j . Если проехать из i в j невозможно, выведите -1 .

Примеры

roads.in	roads.out
5 4 6	0
1 2	-1
2 3	1
1 3	2
4 5	
? 1 2	
? 1 5	
- 2 3	
? 2 3	
+ 2 4	
? 1 5	