

Задача А. МишаСвета

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

Миша и Света поступили в Угабуга Политех™ и, как настоящие студенты, поселились в общежитие №52. Поскольку они были близкими друзьями, администрация общежития разрешила поселиться им в соседних комнатах, прямо через стену. На радостях ребята вбежали в свои комнаты и обомлели — прямо в стене, которая их разделяла, торчал длинный стальной столб. Мише и Свете не понравился такой дизайн интерьера, поэтому они достали из своих чемоданов комично большие молоты и стали бить по этому столбу, пытаясь полностью вбить его в стену.

Поскольку ребята находились с разных сторон этой стены, между ними началось соревнование — кто кому быстрее и на сколько сильно загонит стальной столб. Посчитайте, кто станет победителем в этом сражении и насколько сильно он задоминировал над проигравшим.

Формат входных данных

В первой строке дано одно число n — общее количество ударов студентов. ($1 \leq n \leq 10^5$)

В последующих n строках находится информация о каждом ударе по столбу: первая буква имени ударяющего "M" и "S" для Миши и Светы соответственно, и сила удара k , выраженная в расстоянии, насколько получилось сдвинуть столб за удар. ($1 \leq k \leq 10^9$)

Формат выходных данных

В одной строке выведите первую букву имени победившего и насколько он смог сдвинуть столб от себя, относительно начального положения столба. Если столб по итогу никуда не сдвинулся, выведите 0.

Пример

стандартный ввод	стандартный вывод
5 M 5 M 10 S 3 M 10 S 3	M 19

Задача В. Карточный домик

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

Есть карточный домик высоты n . Карточный домик высоты 1 представляет из себя две карты, опирающиеся друг на друга. Карточный домик высоты $h > 1$ представляет из себя конструкцию, в которой в ряд выстроены h домиков высоты 1, сверху на них горизонтально положили $h - 1$ карту, а сверху на эту конструкцию домик высотой $h - 1$. Из домика выбивают пару карт, представляющую собой домик высоты 1. Каждая такая пара задаётся парой чисел. Первое число — это номер слоя, где нижний — первый. Второе число — это номер пары в слое, нумеруя слева направо. Из-за этого часть карт осталась без опоры и упала, из-за них упали другие карты и так далее. Ваша задача сказать, сколько суммарно карт упало, включая выбитые.

Формат входных данных

В единственной строке входного файла находятся числа $n(1 \leq n \leq 10^8)$ — количество слоев, $l(1 \leq l \leq n)$ — номер слоя, $p(1 \leq p \leq n - l + 1)$ — номер позиции в слое.

Формат выходных данных

Необходимо вывести единственное число — сколько карт суммарно упадёт.

Примеры

стандартный ввод	стандартный вывод
3 2 2	5
3 1 3	8

Замечание

На следующих изображениях приведены иллюстрации к примерам из условия. Красным выделены карты, которые выбиваются, серым — те, которые упадут, а чёрным — те, которые останутся.

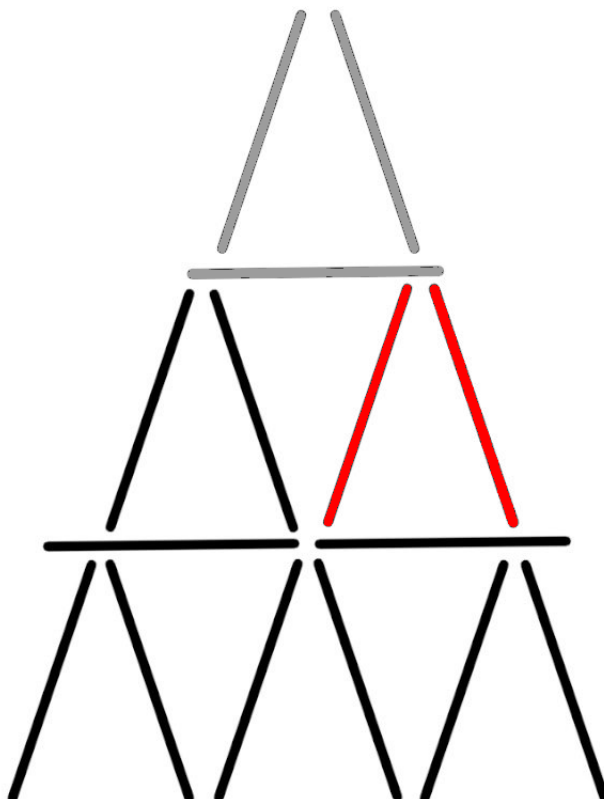


Иллюстрация к примеру 1

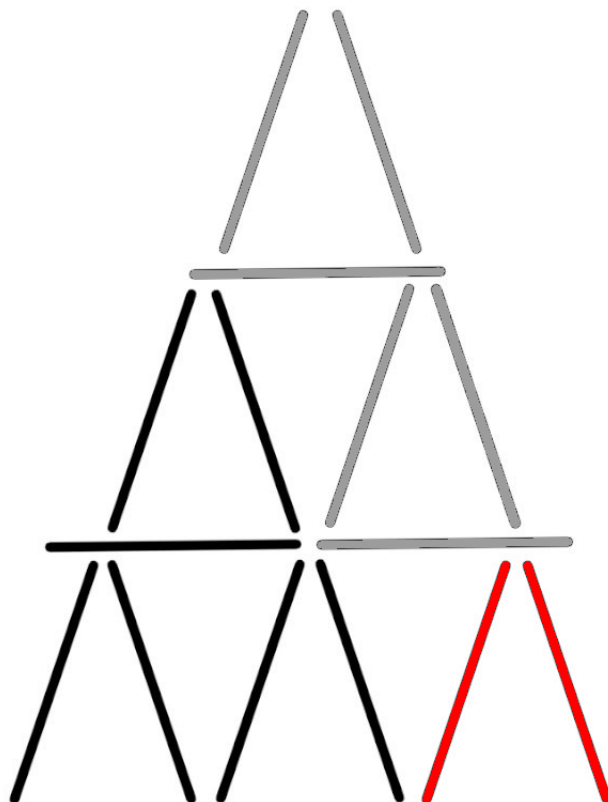


Иллюстрация к примеру 2

Задача С. Трамвайные остановки

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

В Ростове-на-Дону существует уникальная система трамвайных маршрутов. Каждая остановка кодируется строкой из n двоичных цифр, где каждая цифра указывает наличие определенного сервиса на остановке (1 — есть, 0 — нет).

Трамвай может проехать между двумя остановками, если их коды отличаются ровно в одной позиции. Это означает, что маршрут проходит только между «соседними» остановками.

Стоимость проезда до остановки равна сумме тарифов за использование сервисов на этой остановке. Если i -я цифра в коде остановки равна 1, то пассажир должен оплатить i -й тариф.

Нужно найти минимальную стоимость проезда от начальной остановки s до конечной остановки e .

Формат входных данных

Первая строка содержит три значения, разделенных пробелами: n — количество сервисов ($1 \leq n \leq 1000$), s — код начальной остановки (строка из n символов '0' и '1'), e — код конечной остановки (строка из n символов '0' и '1').

Вторая строка содержит n целых чисел $c_1, c_2, \dots, c_n$ — стоимости тарифов для каждого сервиса ($1 \leq c_i \leq 10^6$).

Формат выходных данных

Выведите минимальную стоимость проезда.

Примеры

стандартный ввод	стандартный вывод
3 101 011 1 2 3	8
5 00000 11111 5 4 3 2 1	35
4 1111 1000 200 20 20 20	320

Задача D. Хорошие числа, очередные

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

В древнем цифровом храме хранятся таинственные коды — последовательности цифр, в которых заключена сила. Каждый хранитель следит, чтобы его любая трехсимвольная часть кода была **хорошей** — ведь только так энергия чисел остаётся стабильной.

Трёхсимвольные коды (x, y, z) называются хорошими, если:

- сумма цифр в коде не меньше десяти;
- произведение цифр делится на 24.

Великий жрец хочет составить код длины n , где каждые три соседние цифры образуют хорошую тройку. Только такие коды считаются священными и могут быть выгравированы на стенах храма. Первая цифра, разумеется, не может быть нулём — священные тексты не терпят пустоты в начале.

Твоя задача — узнать, сколько существует кодов длины n .

Формат входных данных

В единственной строке входного файла находится целое число n ($3 \leq n \leq 5 \cdot 10^5$) — длина числа.

Формат выходных данных

Необходимо вывести количество чисел длины n , у которых каждые три подряд идущие цифры образуют хорошее число. Ответ необходимо вывести по модулю $10^9 + 7$, так как он может быть слишком большим.

Примеры

стандартный ввод	стандартный вывод
3	277
215	929081963

Задача Е. Ростовские перестановки

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

Городская администрация Ростова-на-Дону решила упорядочить список достопримечательностей вдоль набережной реки Дон. Каждой достопримечательности соответствует уникальный номер (её рейтинг), и их нужно расположить в порядке возрастания.

Однако перестановка достопримечательностей — дело непростое! Можно менять местами только те пары объектов, между которыми разрешено техническое перемещение. Эти разрешённые пары определяются схемой городских коммуникаций.

Более формально, у вас есть массив a из n различных целых чисел (рейтинги достопримечательностей). Вы можете выполнять операцию обмена $\text{swap}(a_i, a_j)$ только если пара индексов (i, j) присутствует в списке разрешённых пар.

Определите, можно ли отсортировать массив по возрастанию, используя только разрешённые обмены. Количество операций при этом может быть любым.

Формат входных данных

Первая строка содержит целое число n ($1 \leq n \leq 2000$) — количество достопримечательностей.

Вторая строка содержит n различных целых чисел $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^9$) — рейтинги достопримечательностей.

Третья строка содержит целое число m ($0 \leq m \leq \frac{n \cdot (n-1)}{2}$) — количество разрешённых пар для обмена.

Следующие m строк описывают разрешённые пары. Каждая строка содержит два целых числа u_i, v_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$), означающих, что можно менять местами достопримечательности на позициях u_i и v_i .

Пары неориентированы: пара (u, v) эквивалентна паре (v, u) .

Формат выходных данных

Выведите YES, если достопримечательности можно отсортировать по возрастанию рейтинга, используя разрешённые обмены. В противном случае выведите NO.

Если возможно отсортировать, то выведите на следующей строке число q ($q \leq 4 \cdot 10^6$) — количество операций, а на следующей строке q чисел — индексы операций обменов в порядке применения их к массиву. Каждая операция задаётся одним числом — номером разрешённой пары из входных данных. Разрешённые пары нумеруются с единицы в порядке их появления во входных данных.

Примеры

стандартный ввод	стандартный вывод
5 3 1 2 5 4 3 1 2 2 3 4 5	YES 3 1 2 3
3 1 2 3 1 2 1	YES 0

Задача F. Перемодулирование

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

Вам дана перестановка $p_1, p_2, \dots, p_n$: массив, где каждое целое число от 1 до n встречается ровно один раз.

Вам доступна следующая операция: выбрать индекс $1 \leq i \leq n - 1$ и сделать $p_i := p_i \bmod p_{i+1}$. Так как деление на 0, запрещено, необходимо, чтобы p_{i+1} не было равно нулю. Здесь $\bmod$ означает деление с остатком. В большинстве языков программирования эта операция обозначается символом $\%$.

Вам необходимо найти любую валидную последовательность, из не более чем n операций, такую, что после её выполнения, в массиве p найдётся **хотя бы** один элемент равный 0, или сообщить, что такой последовательности операций нет. Обратите внимание, что **не нужно** минимизировать число операций.

Формат входных данных

Первая строка входных данных содержит целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных в тесте.

Далее следуют описания наборов входных данных.

Первая строка описания каждого набора входных данных содержит целое число n ($2 \leq n \leq 2 \cdot 10^5$) — длина массива.

Вторая строка набора входных данных содержит n целых чисел $p_1, p_2, \dots, p_n$ ($1 \leq p_i \leq n$) — перестановка p . Гарантируется, что каждое число от 1 до n встречается в этом массиве ровно один раз.

Гарантируется, что сумма n по всем наборам не превышает $2 \cdot 10^5$.

Формат выходных данных

Для каждого набора входных данных:

Если подходящей последовательности операций не существует, выведите -1 . Иначе, в первой строке, выведите число операций $1 \leq k \leq n$. В последующих k строках выведите параметры операций: единственное число $1 \leq i \leq n - 1$ для каждой операции, в порядке выполнения. Если в процессе, для одной из операций $p_{i+1} = 0$, ваше решение будет считаться неверным, так как деление на ноль запрещено. После всех операций, необходимо, чтобы массив p содержал хотя бы один 0.

Вам не нужно минимизировать число операций. Если решений несколько, можно вывести любое из них.

Пример

стандартный ввод	стандартный вывод
4	2
3	2
1 3 2	1
4	1
4 1 2 3	1
2	-1
1 2	6
9	6
1 2 3 9 7 5 6 4 8	4
	7
	3
	8
	2

Задача G. Побег Валеры

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	256 мегабайт

Поле, по которому путешествует хомяк Валера, представляет собой прямоугольную сетку размером w на h узлов. В некоторых узлах этого поля сидят голодные лисы, которые не прочь полакомиться хомяком.

Валера стартует в узле (x_1, y_1) и должен добраться до узла (x_2, y_2) . Он может перемещаться только между соседними по стороне узлами (т.е. совершать переходы вверх, вниз, влево или вправо).

Чтобы остаться в живых, Валера должен планировать свой маршрут с максимальной осторожностью. **Осторожностью маршрута** назовем минимальное из евклидовых расстояний от всех узлов этого маршрута до ближайшей лисы. Евклидово расстояние между узлами (x_1, y_1) и (x_2, y_2) вычисляется как $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$.

Ваша задача — найти маршрут с **максимально возможной осторожностью**. Искомой величиной является квадрат этой максимальной осторожности.

Формат входных данных

Первая строка содержит два целых числа w и h ($2 \leq w, h \leq 1500$) — ширина и высота поля.

Вторая строка содержит четыре целых числа x_1, y_1, x_2, y_2 ($1 \leq x_1, x_2 \leq w, 1 \leq y_1, y_2 \leq h$) через пробел — координаты старта (x_1, y_1) и финиша (x_2, y_2) соответственно.

Третья строка содержит одно целое число n — количество лис на поле ($1 \leq n \leq \min(w \cdot h, 10^6)$).

Следующие n строк содержат по два целых числа x_i, y_i через пробел — координаты узла, в котором сидит i -я лиса.

Гарантируется, что:

- Все лисы находятся в разных узлах
- Ни одна лиса не находится в стартовом узле или финишном узле

Формат выходных данных

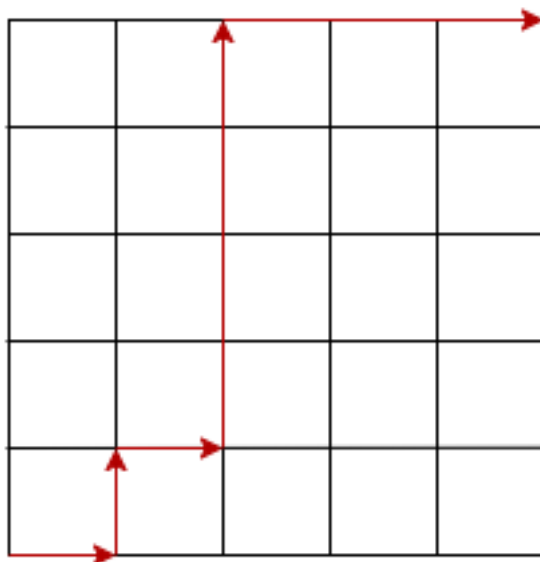
Выведите одно целое число — квадрат максимальной осторожности маршрута, который может выбрать Валера. Если не существует ни одного маршрута, который бы не проходил через узлы с лисами, выведите 0.

Пример

стандартный ввод	стандартный вывод
6 6 1 1 6 6 2 1 3 4 1	2

Замечание

Путь выбранный Валерой для первого примера.



Задача Н. Никита и рыбалка

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

После длительного отдыха Никита решил сходить на рыбалку. Он отправился в ближайший магазин Шестёрочки и купил удочку с леской длины k . Затем, по совету знакомого, Никита отправился на озеро, которое после продолжительных дождей можно представить в виде числовой прямой.

На некоторых координатах этой прямой находятся рыбки. Никита может выбрать только одну координату, на которой поставит лодку, и один раз закинуть удочку. Все рыбки, оказавшиеся между положением лодки и концом закинутой лески, будут пойманы. Более формально, если рыбак находится в координате x , то он поймает всех рыбок на одном из двух отрезков: $[x, x + k]$ или $[x - k, x]$.

Определите, какое максимальное количество рыбок сможет поймать Никита.

Формат входных данных

В первой строке вводятся числа n и k — количество рыбок и длина лески ($1 \leq n \leq 10^5$), ($1 \leq k \leq 10^9$).

Далее идут n чисел a_i — координаты рыбок ($-10^9 \leq a_i \leq 10^9$).

Формат выходных данных

Выведите единственное значение — максимальное количество рыбок, которые заберет Никитос.

Примеры

стандартный ввод	стандартный вывод
5 3 1 2 4 5 7	3
4 1 -1 -3 2 4	1

Задача I. Бюро находок

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

После того как были приведены в порядок автобусные маршруты, мэр решил привнести в город немного современности, построив скоростной трамвай в самом западном районе Ростова-на-Дону — Левенцовке. Поскольку большого опыта в этом нет, было решено, что линия трамвая будет прямой. Таким образом, повороты не потребуются, и строительство инфраструктуры станет проще.

Однако возникла новая проблема: где разместить бюро находок. Этот вопрос оказался куда важнее, чем кажется на первый взгляд. Люди постоянно теряют вещи, садясь и выходя из трамвая: покупки, рюкзаки, ключи, телефоны... В итоге множество пассажиров что-то оставляет, и им приходится искать свои вещи.

Мэр понимает, что должен заботиться о своих горожанах. Поэтому он решил разместить бюро находок на трамвайной остановке, которая находится «ближе» всего к остальным. Однако это не так просто. Некоторые остановки гораздо более популярны, чем другие. Если разместить бюро на малолюдной станции, возможно, она окажется ближе к остальным, но менее удобной из-за низкой посещаемости; другая же, наоборот, может быть немного дальше, но благодаря большому пассажиропотоку будет более подходящей.

Лучший стажёр-урбанист, которого когда-либо видела эта городская администрация, предложил логичный подход: каждой остановке присваивается некоторая *популярность* p_i . Тогда остановка i , на которой следует разместить бюро находок, должна быть такой, что сумма расстояний до всех остальных остановок j , умноженных на их популярность p_j , минимальна. Таким образом, бюро окажется как можно ближе к наиболее популярным станциям.

Формально мы хотим найти такую станцию i , которая минимизирует выражение:

$$\sum_{j \neq i} p_j |a_i - a_j|$$

Но какая же это станция? Помогите урбанисту найти ответ.

Формат входных данных

В первой строке дано число n — количество станций на линии ($2 \leq n \leq 3 \cdot 10^5$).

Во второй строке заданы n целых чисел $a_0, a_1, \dots, a_{n-1}$ — координаты точек на прямой, в которых расположены станции. Гарантируется, что никакие две станции не находятся в одной и той же точке, и что координаты заданы в порядке возрастания.

В последней строке даны ещё n целых чисел $p_0, p_1, \dots, p_{n-1}$, обозначающих популярность каждой станции.

Все числа a_i и p_i — натуральные, не превышающие 10^6 .

Формат выходных данных

Для каждой запланированной линии трамвая требуется вывести индекс станции, на которой следует разместить бюро находок. Если существует несколько станций, минимизирующих суммарное расстояние до остальных станций, выбирается станция с наименьшим индексом.

Примеры

стандартный ввод	стандартный вывод
4 1 3 5 12 8 3 4 1	1
5 1 8 12 14 15 2 2 2 2 2	3

Замечание

Символ $\sum$ называется *сумматор* (знак суммы). Условие $j \neq i$ означает, что необходимо суммировать все слагаемые, кроме того, где $j = i$. То есть:

$$\sum_{j \neq i} p_j |a_i - a_j| = p_0 |a_i - a_0| + p_1 |a_i - a_1| + \dots + p_{i-1} |a_i - a_{i-1}| + p_{i+1} |a_i - a_{i+1}| + \dots + p_n |a_i - a_{n-1}|$$

Пояснение для первого примера:

- Если разместить бюро находок на станции 1, то расстояние будет равно $3 \cdot 2 + 4 \cdot 4 + 1 \cdot 11 = 33$.
- Если разместить его на станции 2, то расстояние будет равно $8 \cdot 2 + 4 \cdot 2 + 1 \cdot 9 = 33$.
- Если разместить его на станции 3, то расстояние будет равно $8 \cdot 4 + 3 \cdot 2 + 1 \cdot 7 = 45$.
- Если разместить его на станции 4, то расстояние будет равно $8 \cdot 11 + 3 \cdot 9 + 4 \cdot 7 = 143$.

Так как по условию при равенстве значений выбирается станция с меньшим индексом, бюро находок следует разместить на станции 1.

Задача К. Гигантские Буквы

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

На въезде в город Ростов-на-Дону установлена знаменитая Тачанка с гигантскими буквами, образующими название города. Каждая буква имеет свою **высоту**. Местные жители, большие поклонники олимпиадной информатики, любят исследовать свойства последовательностей, скрывающихся в окружающем их городском пейзаже.

Рассмотрим непустую последовательность положительных целых чисел $A = (a_1, a_2, \dots, a_N)$, где каждое a_i соответствует высоте одной из букв на стене.

Для любых индексов i и j таких, что $1 \leq i \leq j \leq N$, введём следующие определения:

- $\text{НОД}(i, j) = \gcd(a_i, a_{i+1}, \dots, a_j)$,
- $\text{Макс}(i, j) = \max(a_i, a_{i+1}, \dots, a_j)$,
- $\text{Произв}(i, j) = \text{НОД}(i, j) \cdot \text{Макс}(i, j)$.

Функция $\gcd$ обозначает **наибольший общий делитель** (НОД) последовательности чисел.

Искомая величина:

$$F(A) = \sum_{1 \leq i \leq j \leq N} \text{Произв}(i, j)$$

Требуется вычислить значение $F(A)$ по модулю 1 000 000 007.

Формат входных данных

Первая строка содержит целое число N ($1 \leq N \leq 2 \cdot 10^5$) — количество букв.

Вторая строка содержит N целых чисел $a_1, a_2, \dots, a_N$ ($1 \leq a_i \leq 10^9$) — высоты букв.

Формат выходных данных

Выведите одно число — значение $F(A)$ по модулю 1 000 000 007.

Примеры

стандартный ввод	стандартный вывод
1 14	196
2 11 16	393
3 20 11 12	717