

Задача А. Капли

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

Странствуя по загадочным измерениям, Рик обнаружил одно совершенно уникальное, где били ключи с редчайшим топливом, требующимся ему для конструирования нового изобретения. Чтобы отыскать эти ключи, Рик, естественно, решил отправиться туда вместе с внуком.

После долгих блужданий по измерению, путешественники обнаружили, что нет там никаких ключей, и топливо стекает расположенными в ряд n маленькими каплями с горизонтально подвешенной в воздухе трубы.

Рик и Морти заметили, что каждая капля падает вниз с какой-то своей периодичностью, а именно — каждые p_i секунд, а также то, что каждые k секунд внешняя поверхность трубы очищается и каждая капля начинает расти сначала, причем если капля готова упасть в момент очистки трубы, она падает, и только после этого происходит очистка.

У Рика и Морти есть расширяющаяся до произвольных размеров емкость для сбора жидкости, то есть они могут собрать каждую упавшую каплю, но на это у них есть всего t секунд, после этого их могут заметить.

Помогите героям посчитать количество капель, которые они смогут собрать, если они подошли к трубе сразу после ее очистки и тут же приступили к сбору.

Формат входных данных

Первая строка входных данных содержит три натуральных числа n , m и k — количество капель, периодичность очистки трубы и имеющееся у героев время для сбора топлива, соответственно ($1 \leq n \leq 10^5$, $1 \leq k \leq 10^9$, $0 \leq t \leq 10^9$).

Во второй строке находятся n целых чисел p_i , задающих периодичность падения каждой капли ($1 \leq p_i \leq 10^9$).

Формат выходных данных

Выведите одно число — количество капель, которое упадет с трубы за имеющееся у Рика и Морти время.

Пример

стандартный ввод	стандартный вывод
3 5 17 1 2 3	27

Задача В. Газорпазорп

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

Это интерактивная задача.

В очередной раз путешествуя между мирами, Рик и Морти зашли в магазин антиквариата, где Морти выпросил у Рика купить ему куклу-робота. Как оказалось, эта кукла была не обычной и могла производить на свет потомство — маленьких газорпазорпов, что, собственно, и произошло. Чтобы разобраться, что это за кукла и что делать, Рик отправился на планету происхождения куклы — Газорпазорп. Там его ждали шестирукие женщины-амазонки, которые не признавали мужчин. Так Рик попал в плен. Однако, используя свое обаяние, он смог договориться с женщиной-вождем об условиях освобождения — если он выиграет у нее партию в «Вычитпачисл», он может быть свободен с условием, что больше никогда не появится на их планете. Однако, надо все-таки еще выиграть партию в эту игру.

Правила игры «Вычитпачисл» довольно просты: в начале первый игрок называет три числа x_0 , m и k , после этого каждый игрок на своем ходе должен назвать любое натуральное число x_i , такое, что:

- x_i не больше предыдущего названного числа x_{i-1} , но при этом не меньше $x_{i-1} - m$;
- В течение игры число x_i не может быть названо k или более раз подряд.

Например, если $x_{i-1} = 2$, $k = 3$ и $m > 0$, а также число 2 было уже названо 2 раза, то единственный возможный ход — назвать число 1.

Игрок, который не может сделать ход, вместо числа должен сказать «I'm giving up», после чего он проигрывает.

Первым игроком (то есть тем, кто называет первых 3 числа), конечно же, является вождь. Рик знает, что игра странная и все можно подстроить так, что он проиграет, однако он также знает, что женщины с этой планеты не очень умны, поэтому у него есть шанс выиграть. Помогите ему выиграть в этой игре, а если это сделать все-таки невозможно, скажите ему об этом, чтобы он продумал план побега!

Протокол взаимодействия

Во время взаимодействия вашей программы с программой жюри происходит следующее: сначала на вход вашей программе дается три числа x_0 , m и k ($1 \leq x_0 \leq 1\,000$, $0 \leq m \leq 1\,000$, $2 \leq k \leq 10$), а затем несколько раз повторяются следующие действия:

- ваша программа сообщает программе жюри натуральное число x_i — очередной ход или строку «I'm giving up», если вы решаете сдаться. В случае, если вы решили сдаться, ваша программа должна быть немедленно завершена.
- программа жюри сообщает вашей программе свой ход — натуральное число x_i или «I'm giving up», если программа жюри сдается. Во втором случае, ваша программа должна немедленно завершиться.

Замечание

После каждого действия вашей программы выводите символ перевода строки. Если вы используете «writeln» в Паскале, «cout << ... << endl» в C++, «System.out.println» в Java или «print» в Python, сброс потока вывода у вас происходит автоматически, дополнительно делать «flush» не обязательно. Если вы используете другой способ вывода, рекомендуется делать «flush», но все равно обязательно требуется выводить символ перевода строки.

Ниже приведены наиболее типичные причины получения тех или иных сообщений об ошибке.

Если ваша программа соблюдает протокол, но в итоге сдается, хотя изначально можно было выиграть, вы получите вердикт «Wrong Answer».

Если ваша программа выводит некорректно отформатированные сообщения программе жюри, то вы получите результат «Presentation Error» либо «Wrong Answer».

Если ваша программа нарушила протокол и ждет ввода в то же время, когда его ждет и программа жюри, то вы получите результат «Idleness Limit Exceeded». Обратите внимание, что к такому же результату может привести и то, что вы не переводите строку после каждого выведенного сообщения или выводите не тем способом, который описан в начале раздела, и не делаете «flush».

Примеры

стандартный ввод	стандартный вывод
5 3 3 4 2 1 I'm giving up	4 3 2 1
5 3 2 1	4 I'm giving up

Замечание

В первом тестовом примере представлено одно из развитий событий, где Рик выигрывает.

Во втором тестовом примере Рик не может выиграть при правильной игре первого игрока, поэтому итоговый проигрыш будет также засчитан как правильный ответ. Обратите внимание, что Рик не может назвать число 5, так как в таком случае, оно будет названо два раза подряд.

Задача С. Морти и пароль

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

Однажды Рик решил проверить, насколько смекалист его внук Морти. Как известно, человек лучше думает в экстремальных ситуациях, поэтому Рик запер любимую Морти, Джессику, в комнате и повесил на дверь кодовый замок. Для того, чтобы ее спасти, Морти нужно провести эксперимент, который придумал Рик.

Рик выставил в ряд перед Морти n стаканчиков с соком, на каждом из которых было написано целое число от 1 до n . Число на i -м слева стаканчике было равно a_i . Кроме того, оказалось, что все числа на стаканчиках различны, то есть образовывали перестановку чисел от 1 до n .

Рик разрешил Морти сколько угодно раз брать два соседних стаканчика и менять их местами. Морти очень боится, что никогда снова не увидит свою возлюбленную, поэтому у него трясутся руки, и, когда он меняет местами два стаканчика, из обоих стаканов выливается часть сока. Рик не хочет, чтобы Морти расплескал слишком много сока, поэтому он разрешил внуку прикасаться к каждому стаканчику не более двух раз. Паролем от сейфа является лексикографически максимальная перестановка чисел на стаканчиках, если смотреть слева направо, которая может получиться в результате эксперимента.

Помогите Морти найти пароль и спасти Джессику!

Формат входных данных

В первой строке задано целое число n ($1 \leq n \leq 100\,000$) — количество стаканчиков.

Во второй строке задано n целых чисел $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq n$) — числа на стаканчиках вначале эксперимента. Гарантируется, что все a_i различны.

Формат выходных данных

Выведите n целых чисел через пробел — пароль от сейфа.

Примеры

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

Замечание

Перестановка a длины n лексикографически больше перестановки b длины n , если существует такое x , что $a_i = b_i$ для всех i от 1 до $x - 1$ и $a_x > b_x$.

Задача D. Матрица Рика

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

Рик закончил работу над своим новым изобретением — матрицей $n \times m$, в каждой клетке которой записано натуральное число. К сожалению, оно не заработало, и Рик его попросту выбросил.

На следующий день его нашёл Морти. Он, разумеется, не понял, что это такое, но захотел сделать так, чтобы каждая строка и столбец этого изобретения стали палиндромами.

Какое минимальное число элементов матрицы ему понадобится для этого изменить?

Формат входных данных

В первой строке входных данных заданы числа n и m — количество строк и столбцов в матрице ($1 \leq n, m \leq 1000$).

В следующих n строках находится по m чисел — элементы матрицы, целые положительные числа, не превышающие 10^6 .

Формат выходных данных

Выведите одно число — минимальное количество элементов матрицы, которые нужно изменить, чтобы все строки и столбцы стали палиндромами.

Примеры

стандартный ввод	стандартный вывод
2 4 1 2 2 1 1 2 9 9	2
3 3 1 2 1 1 2 1 1 3 1	1

Замечание

В первом примере можно получить следующую матрицу:

1 2 2 1
1 2 2 1

Во втором:

1 2 1
1 2 1
1 2 1

Задача Е. Кошелёк

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

Рик превратил себя в кошелёк! Бум! Что вы на это скажете? Вот так поворот! Рик — кошелёк. Что вы на это скажете? Этого не происходило ещё никогда. Рик превратил себя в кошелёк! **КОШЕЛЁК-РИК!**

Так как Рик теперь не может двигаться, Морти вынужден помогать ему с обратным превращением. Для этого нужно правильным образом добавить в кошелёк несколько купюр. Всего есть n типов купюр, пронумерованных числами от 1 до n так, что чем больше номер купюры, тем больше ее номинал. В кошельке уже лежит некоторая сумма денег: для каждого i от 1 до n в кошельке уже лежит a_i купюр этого вида. Более того, купюры лежат там в ряд и отсортированы по номиналу, то есть если в кошельке лежат две купюры типов i и j , где $i < j$, то купюра типа i расположена левее купюры типа j . Это свойство должно сохраняться в каждый момент: по словам Рика, если его нарушить, последствия будут непредсказуемы!

Благо у Морти есть две изначально пустые стопки s_l и s_r , в которые он в процесс своих действий может класть купюры.

Кошелёк поддерживает 4 вида операций:

1. Вынуть из кошелька самую левую купюру и положить её с правого края стопки s_l .
2. Вынуть из кошелька самую правую купюру и положить её с левого края стопки s_r .
3. Вставить с левого края кошелька новую купюру.
4. Вставить с правого края кошелька новую купюру.

После каждой операции третьего или четвёртого типа необходимо тут же возвратить все вынутые купюры в кошелёк, а именно — все купюры стопки s_l , не меняя их расположения, поместить в левый край кошелька, а s_r — в правый край. В самом начале стопки s_l и s_r пустые.

Чтобы превратить Рика из кошелька назад в человека, Морти должен по очереди поместить в кошелёк m новых купюр, j -я добавленная купюра должна быть типа b_j . Купюры должны быть добавлены операциями 3-го и 4-го видов. После каждой операции купюры должны оставаться отсортированными по номиналу слева направо. Рик и Морти, чтобы понять, сколько времени у них займёт этот процесс, обратились к вам за помощью. Напишите программу, которая определит, какое наименьшее число операций первого и второго типов потребуется непосредственно перед каждым добавлением очередной купюры.

Формат входных данных

Первая строка входных данных содержит число n — количество различных типов купюр ($1 \leq n \leq 10^5$).

Вторая строчка содержит n целых чисел a_i — изначальное количество купюр каждого типа, которые лежат в кошельке ($0 \leq a_i \leq 10^5$).

В третьей строке находится число m — количество купюр, которые надо добавить в кошелёк, чтобы Рик обратно превратился в человека ($1 \leq m \leq 2 \cdot 10^5$).

В последней строке входных данных находится m чисел b_j — типы купюр, которые надо добавить в кошелёк, что бы Рик снова стал человеком ($1 \leq b_j \leq n$).

Формат выходных данных

В первой строке выходных данных выведите через пробел количество операций первого и второго типа, которое надо будет совершить перед каждым добавлением очередной купюры в кошелёк.

Пример

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

Замечание

Будем изображать текущее положение в формате $s_l/s/s_r$, где s — последовательность купюр, лежащих в кошельке.

В первом тесте в самом начале ситуация выглядит как $/1122233/$, после чего осуществляется 5 действий:

1. Чтобы добавить купюру типа 2, надо с любой из сторон дважды взять купюру, потом с той же стороны засунуть купюру типа 2.
2. В $/1122233/$ добавить купюру типа 1 можно, просто засунув эту купюру с левой стороны.
3. Из $/11122233/$ быстрее всего вынуть два раза купюру справа, получить $/111222/33$ и добавить справа купюру типа 2.
4. В $/111222233/$ сразу добавляем справа купюру типа 3.
5. Нет разницы, с какой стороны в $/1112222333/$ три раза вынуть купюру. Например, слева: получим $111/2222333/$ и добавим слева купюру типа 2, получив финальную ситуацию $/1112222333/$.

Задача F. Безопасное путешествие

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

Однажды во сне Рик увидел вселенную, состоящую из нескольких планет. Между некоторыми из них существуют пути (не более одного между двумя планетами), каждым из которых можно воспользоваться для перемещения между двумя определенными планетами в любую сторону. Рик понял, что хочет начать свое путешествие на случайной планете p , затем воспользоваться случайным путем перемещения на другую планету, и так далее. При этом Рик не интересно посещать какую-либо планету дважды, поэтому на каждом шаге следующая планета будет выбираться только среди непосещенных. Когда на очередном шаге Рик не сможет переместиться на непосещенную планету, он переместится на исходную планету p (если, конечно, существует путь на планету p с текущей планеты).

Рик сразу заметил, что вселенная является *безопасной*, то есть при такой стратегии путешествия он всегда побывает на всех n планетах, и закончит свое путешествие на исходной планете p .

К сожалению, после пробуждения Рик не смог вспомнить, между какими планетами вселенной существуют пути. Однако, он помнит, что во вселенной было n планет и m путей между ними.

Помогите Рикку придумать любую безопасную вселенную, состоящую из ровно n планет и ровно m путей между ними, или определите, что это невозможно.

Формат входных данных

Первая и единственная строка входных данных содержит два натуральных числа n и m — количества планет и путей перемещения между ними ($1 < n \leq 13$, $0 \leq m \leq \frac{n \cdot (n-1)}{2}$).

Формат выходных данных

Если хотя бы одна требуемая вселенная существует, выведите на первой строке **Possible**, затем m строк, i -я из которых содержит два натуральных числа a_i и b_i ($0 < a_i, b_i \leq n$, все неупорядоченные пары $\{a_i, b_i\}$ различны), разделенных ровно одним пробелом — номера планет, между которыми есть путь перемещения.

Иначе выведите **Impossible**.

Примеры

стандартный ввод	стандартный вывод
3 3	Possible 1 3 1 2 2 3
2 0	Impossible

Задача G. Морти и подпоследовательности

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

Все знают, как Рик и Морти любят путешествовать и влезать в авантюры! И новое путешествие не исключение! Перед тем как отправиться, Рик попросил Морти помочь ему справиться с одной жизненно-важной задачей, без которой путешествию не состояться. Маленький Морти уже попытался справиться, но у него ничего не вышло, именно поэтому он решил обратиться за помощью к вам!

Задача, которую дал ему Рик звучит следующим образом: дан массив a из n целых положительных чисел. Для всех целых k , для которых выполняется неравенство $1 \leq k \leq n$ нужно определить, сколько какое максимальное число элементов можно оставить, убрав некоторые, так, чтобы оставшийся массив можно было разбить на подотрезки, каждый из которых — возрастающая последовательность, длины не меньше k .

Последовательность $a_{i_1}, a_{i_2}, \dots, a_{i_p}$ называется возрастающей подпоследовательностью в массиве a , если $a_{i_1} < a_{i_2} < \dots < a_{i_p}$.

Размер последовательности — количество элементов, которые принадлежат последовательности.

Формат входных данных

Первая строка входных данных содержит одно целое число n ($1 \leq n \leq 300$) отвечающее за длину массива. На второй строке содержится массив a из n целых чисел, $1 \leq a_i \leq 10^9$.

Формат выходных данных

В единственной строке выведите n чисел b_i — максимальное число элементов, которые войдут в непересекающиеся возрастающие подотрезки размера не менее i путем исключения некоторого (возможно нулевого) числа элементов из исходного массива.

Примеры

стандартный ввод	стандартный вывод
3 1 2 3	3 3 3
2 1 1	2 0
5 1 4 3 2 9	5 4 3 0 0

Замечание

Рассмотри третий пример. Для $k = 1$, ответ равен 5, так как каждый элемент по отдельности является возрастающей последовательностью. Для $k = 2$ максимальный ответ достигается путем избавления, например, от числа 4, разбивая оставшийся массив на два отрезка длины 2, которые являются возрастающими последовательностями. Для $k = 3$ максимальный ответ можно достичь удалив элементы со значениями 4 и 3, в результате получив один отрезок, который является возрастающей последовательностью длины 3.

Задача Н. Морти покупает продукты

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	6 секунд
Ограничение по памяти:	256 мегабайт

Рик и Морти пришли в магазин. Как вы уже поняли, Рик очень любит давать Морти разные задания. И этот случай не является исключением! Сейчас ему нужно подсчитать количество способов купить k продуктов. Всего в магазине n продуктов. Стоимость i -го продукта равняется c_i . Рик будет q раз давать Морти два числа — l и r , Морти в свою очередь должен сказать, сколько существует способов купить k продуктов, чтобы их суммарная стоимость была не меньше l и не больше r . Обозначим покупку, как набор индексов $i_1, i_2, \dots, i_k$, где i_j — номер товара, который был куплен j -м. Два способа покупки A и B называются различными, если (индекс товара, купленным d -м будем обозначать как A_d) существует такой индекс d , что $A_d \neq B_d$. Таким образом $(1, 2)$ и $(2, 1)$ это два разных способа покупки товаров. Так же, заметьте, что один продукт можно покупать сколько угодно раз.

Формат входных данных

В первой строке входного файла содержатся три числа n, k, q , ($1 \leq n, q \leq 10^5$), ($1 \leq k \leq 10^5$). Во второй строке находится n целых чисел c_i обозначающих стоимости товаров. Товар с индексом i имеет стоимость c_i . ($1 \leq c_i \leq 5 \cdot 10^4$). Следующие q строк содержат два числа l и r обозначающие вопрос от Рика, ($1 \leq l \leq r \leq 5 \cdot 10^4$). Морти должен сказать, сколько существует способов купить k продуктов, чтобы их суммарная стоимость была не меньше l и не больше r .

Формат выходных данных

В q строках выведите ответы на запросы Рика. Так как ответ может быть очень большим, выведите его по модулю 786433.

Примеры

стандартный ввод	стандартный вывод
5 1 5 1 2 3 4 5 1 2 1 3 1 4 1 5 2 5	2 3 4 5 4
2 2 1 1 1 1 10000	4

Задача I. Портальная пушка

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

Для путешествия между вселенными Рик постоянно использует свою портальную пушку. Однако, как и у всех его изобретений, внутри ее механизма заложено огромное количество формул и вычислений.

Каждая вселенная описывается некоторым набором натуральных чисел. Для перемещения из вселенной A , описываемой набором чисел $a_1, a_2, \dots, a_n$ во вселенную B , описываемую набором чисел $b_1, b_2, \dots, b_m$, сначала необходимо посчитать их гравитационную разницу, равную $\sum_{i,j} (i - j) \cdot |a_i - b_j|$.

Чтобы немного почувствовать себя гением, Рик дал вам задачу посчитать гравитационную разницу двух данных миров A и B , пока он будет тусовать с Морти в Blips and Chitz.

Формат входных данных

В первой строке содержится число n — количество чисел, которыми описывается вселенная A ($1 \leq n \leq 10^5$).

Во второй строке содержится n чисел a_i — числа, описывающие вселенную A ($1 \leq a_i \leq 10^4$).

В третьей и четвертой строках содержится описание вселенной B в том же формате.

Формат выходных данных

В единственной строке выведите гравитационную разницу вселенных A и B .

Примеры

стандартный ввод	стандартный вывод
3 1 2 3 3 1 2 3	0
4 1 4 3 6 3 8 1 1	34

Задача J. Спасите Землю

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

На землю началось инопланетное вторжение. Но все не так безвыходно, как может показаться на первый взгляд. Рик может хакнуть корабли инопланетян и заставить их развернуться и покинуть солнечную систему. Для этого ему нужно подключиться к двум параболическим антеннам, самым большим в округе. Морти даже почти удалось заставить Рика это сделать, но Рик все еще сопротивляется, потому что ему лень.

Чтобы Морти отстал от него на пару минут, после чего Рик все же придется идти спасать мир, Рик попросил Морти найти длину кратчайшего пути, который нужно будет проделать Рiku от дома, где он сейчас находится, чтобы включить обе антенны. Чтобы включить первую антенну, Рiku нужно оказаться от нее на расстоянии не более r_1 , а чтобы включить вторую, на расстоянии не более r_2 .

Как вы могли догадаться, из дома включить ни одну из антенн не получается. А также, в силу технических ограничений, расстояние между антеннами строго больше $r_1 + r_2$.

Считайте, что никаких препятствий на плоскости нет. В том числе, можно идти через то место, где находится антенна.

Формат входных данных

В первой строке даны три целых числа x_1 , y_1 и r_1 — координаты первой антенны на плоскости ($-10^6 \leq x_1, y_1 \leq 10^6$, $1 \leq r_1 \leq 10^6$). Во второй строке даны три целых числа x_2 , y_2 и r_2 — координаты второй антенны на плоскости ($-10^6 \leq x_1, y_1 \leq 10^6$, $1 \leq r_1 \leq 10^6$). В третьей строке даны два целых числа x и y — координаты дома, где сейчас находится Рик ($-10^6 \leq x, y \leq 10^6$).

Формат выходных данных

В единственной строке выведите длину кратчайшего пути, который придется проделать Рiku, чтобы включить обе антенны, с абсолютной или относительной погрешностью не более 10^{-6} .

Примеры

стандартный ввод	стандартный вывод
2 1 1 3 4 1 0 0	4.0000000000
0 0 2 4 -2 1 -2 -4	5.7382161910

Замечание

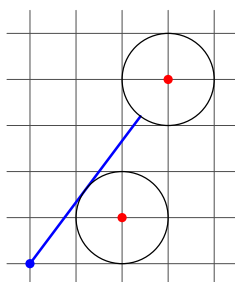


Рис. 1: Пояснение для первого теста

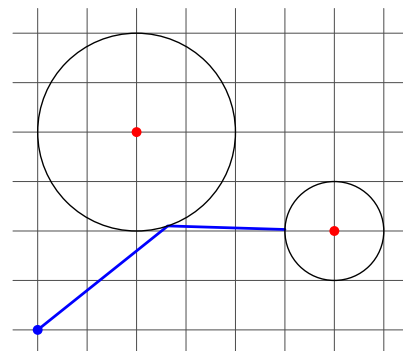


Рис. 2: Пояснение для второго теста