

Задача А. Рисунок

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

Однажды Василий нашёл квадратный шаблон некоторого рисунка, представляющего из себя n строк, каждая из которых имеет длину n , и состоящего из символов «#» и «.».

Оказалось, что у Василия есть электросхема, представляющая из себя также n рядов, в каждой из которых находится n лампочек. Состояния лампочек описываются символами «1» – включенная лампочка и «0» – выключенная.

Лампочки можно переключать только по одной, причём для того чтобы изменить состояние любой лампочки требуется 1 единица времени. Можно менять состояние лампочек, находящихся как в состоянии «1», так и в состоянии «0».

Василий хочет переключить некоторые лампочки таким образом, что бы они *соответствовали* рисунку.

Василий считает, что набор его лампочек *соответствует* рисунку, если для любой пары позиций (i, j) и (k, s) выполнено следующее: если на рисунке символ в позиции i , под номером j , и символ в позиции k , под номером s совпадают, то символы в соответствующих позициях в электро схеме также совпадают, иначе символы в соответствующих позициях в электро схеме различны.

Определите минимальное количество времени, которое нужно Василию, чтобы лампочки в его электросхеме стали *соответствовать* рисунку.

Формат входных данных

Первая строка ввода содержит одно целое число n ($1 \leq n \leq 1000$) – размер таблиц.

Далее следуют n строк, каждая из которых состоит из символов «#» и «.» – описание рисунка.

Затем следуют n строк, каждая из которых состоит из символов «0» и «1» – описание начальных состояний лампочек.

Формат выходных данных

В единственной строке выведи одно число – ответ на задачу.

Система оценки

Правильный ответ на каждом тесте к задаче (не считая тестов из условия) оценивается в 0.4 балла.

Дополнительные ограничения на тесты (не считая тестов из условия):

1-15 тесты – $n \leq 100$.

16-25 тесты – без дополнительных ограничений.

Примеры

стандартный ввод	стандартный вывод
3 ### #.# ### 101 000 001	4
5 ##### #.... ##### ...# ##### 11111 10101 11100 10011 00111	8

Задача В. Бугристый массив

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

В одном небольшом городе, укрытом от посторонних глаз в горах, археологи наткнулись на таинственную пещеру. Внутри неё они обнаружили древние рукописи, написанные на неизвестном языке. После долгих исследований и расшифровки текста учёные пришли к выводу, что записи принадлежат известной цивилизации, существовавшей тысячелетия назад, и содержат сведения о некоторой математической задаче. Они предполагают, что решение этой задачи может оказаться важным для их исследований, и поэтому просят вас написать программу для её решения.

Условие задачи формулируется следующим образом: массив b называется *бугристым*, если каждый его внутренний элемент (то есть не первый и не последний) строго больше или строго меньше обоих его соседей.

Изначально дан некоторый массив a , состоящий из целых чисел. За одну операцию можно выбрать некоторый элемент массива и увеличить или уменьшить его на 1.

Необходимо определить минимальное число операций, которые нужно применить к массиву a , чтобы он стал *бугристым*.

Формат входных данных

Первая строка ввода содержит одно целое число n ($3 \leq n \leq 10^5$) — длина массива a .

Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n ($-10^9 \leq a_i \leq 10^9$) — значения элементов массива a .

Формат выходных данных

Выведите минимальное число операций, которые нужно применить к массиву a , чтобы он стал *бугристым*.

Система оценки

Правильный ответ на каждом тесте к задаче (не считая тестов из условия) оценивается в 0.2 балла.

Дополнительные ограничения на тесты:

1-10 тесты — $n \leq 10^2$.

11-30 тесты — $n \leq 10^3$.

31-50 тесты — без дополнительных ограничений.

Примеры

стандартный ввод	стандартный вывод
5 1 3 5 6 5	3
8 -8 4 -6 5 9 -7 8 1	25

Задача С. Шахматы

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

Недавно Максим узнал правила шахмат, и ему очень понравилась эта игра. Среди всех фигур в шахматах Максим считает, что шахматный конь движется наиболее необычно. Поэтому он придумал для себя упражнение, чтобы лучше видеть, на какие поля может двигаться конь: он расставил шахматные фигуры в некотором порядке на доске и хочет понять, возможно ли за не более чем 3 хода конём добраться до короля противоположного цвета. Если это возможно, он хочет при движении к королю «съесть» фигуры с наибольшим материальным весом в сумме (после взятия короля конь больше не может делать ходы). Максим считает, что пешка соответствует одной единице материала, конь и слон – трём, ладья – пяти, а ферзь – девяти единицам материала.

Помогите Максиму определить, правильно ли он решил задачу. Для этого он просит вас написать программу, выводящую её решение.

Напомним, что конь в шахматах, находясь в точке (x_1, y_1) , может перейти в точку (x_2, y_2) , если $(x_1 - x_2)^2 + (y_1 - y_2)^2 = 5$ и при этом (x_2, y_2) не занято фигурой того же цвета, что и конь.

Также обратите внимание, что расстановка фигур является произвольной и может не быть возможной в реальной партии.

Формат входных данных

Вводится 8 строк, содержащих по 8 символов – описание шахматной доски.

Символ «.» означает пустую клетку, символ «#» означает клетку, занятую фигурой цвета коня, «К» – начальное положение коня, «0» – положение короля противоположного цвета, и «1», «3», «5», «9» – количество единиц материала фигур, которые может взять конь.

Гарантируется, что символы «0» и «К» встречаются в точности 1 раз.

Формат выходных данных

Выведите -1 , если конь не может «съесть» короля, иначе выведите максимальное количество единиц материала, которое возможно взять при движении к королю.

Система оценки

Правильный ответ на каждом тесте к задаче (не считая тестов из условия) оценивается в 0.2 балла.

Примеры

стандартный ввод	стандартный вывод
.#. ##. . 11. ...К... ..9###.. ...5.... ...#.5..... 0..###..	10
.К .#. . .#13 ..# . . .#. ..9.0.#. ...#.9...5.. ..5.#. .#	-1

Задача D. Олимпиады

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

Во второй половине учебного года проходит множество олимпиад, и в одной группе из n человек оказалось, что у каждого есть любимая олимпиада, причём их любимые олимпиады попарно различны. Для удобства будем считать, что у ученика 1 любимая олимпиада имеет номер 1, у ученика 2 любимая олимпиада имеет номер 2 ... и у ученика n любимая олимпиада имеет номер n . Также у каждого ученика есть лучший друг. Для ученика i его лучший друг это ученик p_i . Если $p_i = i$, то это означает, что лучший друг школьника i не входит в данную группу школьников. Оказалось, что массив p является перестановкой, то есть для любых i, j , таких что $i \neq j$, верно $p_i \neq p_j$.

Когда пришло время регистрироваться на олимпиады, каждый школьник решил принять участие в некоторых из них. Оказалось, что школьник i принимает участие в своей любимой олимпиаде, а также во всех олимпиадах, на которые зарегистрировался его лучший друг (при $p_i \neq i$).

Пусть школьник i зарегистрировался на a_i олимпиад. Вам требуется посчитать минимальное значение, которое может принимать сумма чисел массива a , или, другими словами, минимальное число регистраций школьников на олимпиады.

Формат входных данных

В первой строке входных данных содержится единственное целое число n ($1 \leq n \leq 10^5$).

Во второй строке содержатся n чисел p_1, p_2, \dots, p_n ($1 \leq p_i \leq n$). Гарантируется, что p является перестановкой.

Формат выходных данных

В единственной строке выведите одно число – ответ на задачу.

Система оценки

Правильный ответ на каждом тесте к задаче (не считая тестов из условия) оценивается в 0.2 балла.

Дополнительные ограничения на тесты (не считая тестов из условия):

1-10 тесты – $n \leq 10^2$.

11-20 тесты – $n \leq 10^3$.

21-40 тесты – без дополнительных ограничений.

Примеры

стандартный ввод	стандартный вывод
4 4 2 3 1	6
6 4 1 6 2 3 5	18

Задача Е. Эксперимент

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

Группа ученых из МИФИ работает над очередным экспериментом. Они хотят провести эксперимент с различными заряженными частицами.

Во время эксперимента могут происходить события следующих типов:

- 1 $x t$ – появляется новая частица энергии x и типа t .
- 2 $x t$ – любая частица типа t с энергией x исчезает. Если ни одной такой частицы нет, ничего не изменяется.
- 3 $x t$ – энергия всех частиц типа t увеличивается на x единиц.
- 4 – требуется вывести сумму энергий всех частиц.
- 5 – требуется вывести минимальную энергию среди всех частиц, если частиц нет – вывести -1 .

Эксперимент может иметь большое значение для последующей работы, поэтому учёные просят вас написать программу, которая проверит правильность результатов.

Формат входных данных

В первой строке содержится единственное число q ($1 \leq q \leq 3 \cdot 10^5$) – количество запросов. Далее в каждой из последующих q строк содержится очередной запрос в формате, описанном выше.

Гарантируется, что в каждом из них $1 \leq t \leq n$ и $1 \leq x \leq 10^9$ и что все числа в вводе целые.

Формат выходных данных

На каждый запрос типа 4 и 5 выведите требуемое число в отдельной строке.

Система оценки

Правильный ответ на каждом тесте к задаче (не считая тестов из условия) оценивается в 0.2 балла.

Дополнительные ограничения на тесты (не считая тестов из условия):

1-20 тесты – $q \leq 10^3$.

21-30 тесты – $q \leq 10^4$.

31-50 тесты – без дополнительных ограничений.

Примеры

стандартный ввод	стандартный вывод
6 2 10 2 5 1 15 1 1 17 2 3 3 1 4	-1 35
17 1 10 1 1 10 2 3 10 1 5 4 1 20 1 2 20 1 5 2 10 1 4 2 30 1 2 20 1 4 5 2 10 2 4 5	10 30 10 30 10 10 0 -1

Задача А. Рисунок

Из определения *соответствующих* рисунков следует, что либо в итоговых состояниях лампочек «1» соответствует «#», а «0» – «.», либо «0» соответствует «#», а «1» – «. ».

Значит для решения достаточно рассмотреть ответы для обоих вариантов и взять из них минимальный. Для того, чтобы найти ответ для одного из вариантов нужно потратить 1 единицу времени для каждой лампочки, которая не совпадает с соответствующим ей типом.

Задача В. Бугристый массив

Заметим, что если некоторый элемент строго больше своих соседей, то соседние элементы могут быть только строго меньше соответствующих им соседних элементов. Это означает, что если $a_1 < a_2$, то $a_2 > a_3$ и наоборот.

Для решения рассмотрим 2 случая сравнения a_1 и a_2 в итоговом массиве. Пусть $a_1 < a_2$ в итоговом массиве, тогда если это неравенство уже выполнено, не будем изменять a_2 , иначе увеличим a_2 до значения $a_1 + 1$. Это является оптимальной стратегией, т.к. увеличение a_2 только «улучшает» требуемое неравенство $a_2 > a_3$, а значит, если нам необходимо применить некоторое количество операций для достижения $a_1 < a_2$, выгодно применять их к a_2 . После достижения $a_1 < a_2$ увеличение a_2 только «улучшает» $a_2 > a_3$, однако уменьшение a_3 влияет на это неравенство также, но возможно «улучшает» неравенство $a_3 < a_4$, а значит, если нужно выполнить $a_2 > a_3$, не хуже будет применять операции к a_3 . Таким образом, это оптимальное количество операций, которые нужно применить к a_2 .

Случай, когда $a_1 > a_2$, рассматривается аналогично. После применения операций к a_2 рассмотрим неравенство, связывающее a_2 и a_3 , которое будет однозначно определено предыдущим. Выполним этот процесс всего $n - 1$ раз для каждого из двух вариантов сравнения a_1 и a_2 . Получим 2 возможных ответа, из которых выбираем наименьший. Асимптотика решения: $\mathcal{O}(n)$.

Задача С. Шахматы

Для решения задачи запустим рекурсивную функцию из точки, где находится конь. Эта функция будет переходить во все возможные поля, куда может пойти конь (с учётом того, что из поля «0» нельзя никуда ходить) и поддерживать текущее количество съеденного материала, а также запускать себя на не более чем 3 шага. Тогда при достижении клетки достаточно обновлять максимальное количество взятого материала (изначально его можно установить в -1), которое можно собрать, достигнув этой клетки. Ответ будет находиться в клетке «0».

Задача D. Олимпиады

Рассмотрим граф, в котором есть вершины от 1 до n , соответствующие школьникам, в котором рёбра соответствуют лучшим друзьям. В нём из каждой вершины исходит 1 ребро и в каждую вершину входит 1 ребро. Рассмотрим произвольную вершину x , переместимся по ребру, исходящему из неё, затем по ребру из вершины, в которую мы пришли и т.д. В некоторый момент мы придём в ранее посещённую вершину. Она не может быть отличной от x , так как это означало бы, что в неё можно попасть по двум рёбрам. Значит первой совпадающей вершиной будет x . Это верно для любой вершины графа, значит граф разбивается на набор из простых циклов.

Каждый школьник участвует в олимпиаде, на которую зарегистрировался его лучший друг. Покажем, что школьник x должен участвовать во всех олимпиадах из «своего» цикла. Рассмотрим любую олимпиаду из этого цикла. Если кто-либо участвует в ней, то тот ученик, для которого участник является лучшим другом, тоже участвует в ней. Повторив это рассуждение конечное число раз, получим, что все в цикле должны участвовать в олимпиаде любого школьника этого цикла. Заметим, что если это верно для всех циклов, то все условия верны, а значит в этом случае достигается минимум, равный сумме квадратов длин всех циклов. Для решения остаётся посчитать длины циклов. Это можно сделать, создав массив не посещённых чисел от 1 до n и рассмотрев все числа от 1 до n . Когда алгоритм встречает число, которое не было посещено, нужно пройти по соответствующему циклу и найти его длину.

Задача E. Эксперимент

Рассмотрим случай, когда все частицы одного типа. Тогда для быстрого решения задачи можно

хранить все энергии в множестве (`std::multiset` в C++), которое поддерживает добавление, удаление и поиск минимума за $\mathcal{O}(\log n)$, и число d , показывающее, на сколько должны быть увеличены все элементы множества. Тогда операция 3 заключается в увеличении d . Для добавления x нужно добавить $x - d$, и аналогично с удалением. Сумму можно поддерживать отдельно и пересчитывать после каждого запроса.

Тогда для решения задачи будем поддерживать такую структуру для каждого типа частицы, а также сумму всех элементов и множество минимумов. При выполнении операции только одна структура изменяется, значит, можно учитывать только её изменения для пересчёта суммы и множества минимумов. Таким образом, данное решение работает за $\mathcal{O}(q \log q)$.