Задача A. Аборигены

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Капитан Кук с командой попали в плен к аборигенам. Чтобы избежать участи быть съеденными, путешественники предложили аборигенам поделиться с ними сокровищами. Оказалось, что у капитана и его команды есть n сокровищ.

Главный шаман согласен отпустить капитана с командой, если они отдадут им не меньше половины своих сокровищ. Все сокровища выглядят красиво, поэтому шаман согласен на любые сокровища, главное, чтобы их было не меньше половины.

Однако для Кука сокровища представляют определенную ценность. Он считает ценность i-го сокровища равной a_i. Помогите капитану определить, какие сокровища отдать аборигенам, чтобы его с товарищами отпустили, а суммарная ценность оставшихся в распоряжении команды сокровищ была максимальной.

Формат входных данных

На первой строке ввода находится число n ($2 \leqslant n \leqslant 1000$).

На второй строке находятся n натуральных чисел $a_1, a_2, \ldots, a_n$ ($1 \leqslant a_i \leqslant 1000$).

Формат выходных данных

Выведите одно целое число — максимальную суммарную ценность сокровищ, которые могут остаться у капитана Кука и его команды, если они отдадут не менее половины сокровищ аборигенам.

Пример

стандартный ввод	стандартный вывод
6 2 4 1 3 3 5	12

Замечание

В примере Кук может отдать сокровища со стоимостью 1, 2 и 3 шаману, оставив себе сокровища со стоимостью 3, 4 и 5. Их суммарная стоимость равна $3 + 4 + 5 = 12$.

Задача B. Сбалансированная иллюминация

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Правительство Санкт-Битсбурга составляет техническое задание для украшения города к новому году.

Губернатор считает, что следует повесить на главной площади гирлянду из n лампочек. Лампочки будут включаться и выключаться, и гирлянда будет радовать жителей города.

Главный дизайнер города решил, что гирлянда будет менять свой вид каждую секунду. Каждая лампочка, входящая в состав гирлянды, может быть включена или выключена. Дизайнер решил, что раз в секунду ровно одна из лампочек будет менять свое состояние: с включенной на выключенную или наоборот, с выключенной на включенную. При этом дизайнер хочет, чтобы комбинации включенных и выключенных лампочек повторялись с периодом 2^n секунд, причем за время периода в 2^n секунд все возможные 2^n комбинаций состояний лампочек были представлены на гирлянде.

Главный инженер города, однако, отметил, что постоянное включение и выключение лампочек приводит к их износу. Чтобы износ был равномерным, необходимо, чтобы все лампочки включались и выключались примерно одно и то же число раз.

Итоговое техническое задание для вас — главного программиста департамента информационных технологий правительства — выглядит так.

- Необходимо составить план из 2^n конфигураций лампочек $a_0, a_1, \ldots, a_{2^n-1}$, где a_k — строка из n нулей и единиц, $a_k[i] = 1$ означает, что лампочка i в конфигурации a_k включена, а $a_k[i] = 0$ означает, что лампочка i в конфигурации a_k выключена.
- Все конфигурации в плане должны быть различны.
- Этот план будет запущен по циклу, каждую секунду на гирлянде будет показываться очередная конфигурация, в t-ю секунду будет показываться конфигурация $a_{t \bmod 2^n}$.
- Соседние конфигурации должны отличаться состоянием ровно одной лампочки, аналогично конфигурации a_{2^n-1} и a_0 также должны отличаться состоянием ровно одной лампочки.
- Обозначим как c_i количество изменений состояния лампочки i за время полного цикла, включая конечный переход от a_{2^n-1} к a_0. Тогда для любых $i \neq j$ значения c_i и c_j должны различаться не более чем на 2.

За работу!

Формат входных данных

На ввод подается одно целое число n ($1 \leqslant n \leqslant 17$).

Формат выходных данных

Выведите 2^n строк по n символов — искомую последовательность конфигураций в плане. Гарантируется, что подходящий под все условия план существует.

Примеры

стандартный ввод	стандартный вывод
3	000 010 011 111 110 100 101 001
4	0000 0010 1010 1011 0011 0111 0110 0100 0101 0001 1001 1101 1111 1110 1100 1000

Замечание

В первом примере $c_1 = c_2 = 2$, $c_3 = 4$.

Во втором примере $c_1 = c_2 = c_3 = c_4 = 4$.

Задача C. Сколько строк меньше

Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Дан набор D из n строк и строка s. Требуется быстро находить число строк, лексикографически меньших s, в наборе D.

С заданной строкой s выполняются q модификаций, каждая из которых задается парой из числа k_i и символа c_i. Модификация (k_i, c_i) заключается в том, что все символы строки s, начиная с k_i и до конца строки, заменяются на символ c_i.

Например, пусть исходно строка s была равна «anatoly», тогда последовательность запросов $(5, \texttt{o})$, $(3, \texttt{b})$, $(7, \texttt{x})$ будет менять строку следующим образом:

«anatoly» → «anatooo» → «anbbbbb» → «anbbbbx»

После каждого изменения строки s требуется вывести количество строк в наборе D, которые лексикографически меньше, чем s.

Замечание

Строка a лексикографически меньше строки b, если $a \neq b$ и выполнено одно из двух условий:

- a является префиксом строки b;
- для некоторого i первые i символов строки a равны соответствующим символам строки b, а $a_{i+1} < b_{i+1}$.

Формат входных данных

Первая строка входных данных содержит два целых числа n и q — количество строк в наборе D и количество модификаций ($1 \leqslant n, q \leqslant 10^6$).

Во второй строке находится строка s, состоящая из не более чем 10^6 строчных латинских букв.

В следующих n строках содержатся строки набора D. Каждая строка состоит из строчных латинских букв. Суммарная длина строк в D не превосходит 10^6.

Следующие q строк содержат описания модификаций. Описание состоит из числа k_i и строчной буквы латинского алфавита c_i, разделенных пробелом ($1 \leqslant k_i \leqslant |s|$).

Формат выходных данных

В первой строке выведите число строк в наборе D, которые лексикографически меньше изначальной строки s.

Затем выведите q строк. В i-й строке выведите ответ после i-й модификации.

Примеры

стандартный ввод	стандартный вывод
4 3 anatoly boris anatooo anbbbbu anba 5 o 3 b 7 x	0 0 2 3
5 5 abcde buz ababa build a aba 1 b 3 z 2 u 4 z 1 a	3 3 3 4 4 1

Замечание

В первом тесте из примера строка изменяется следующим образом:

«anatoly» → «anatooo» → «anbbbbb» → «anbbbbx».

- Изначальная строка «anatoly» лексикографически меньше всех строк набора, поэтому ответ на задачу 0.
- После первого изменения строка становится «anatooo» и такая строка есть в наборе, однако ответ на задачу по прежнему будет 0, так как она не меньше, текущей.
- Затем строка становится «anbbbbb», что лексикографически больше, чем «anatooo» и «anba», но меньше чем «anbbbbu» и «boris», таким образом ответ 2.
- После последнего изменения строка станет «anbbbbx», что лексикографически больше «anatooo», «anba» и «anbbbbu», ответ 3.

Задача D. Запись на экзамен

Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Обычно Новый год у всех ассоциируется с ёлкой и праздничным столом, однако у студентов Новый год ассоциируется с сессией. Наступило 31 декабря, и студенты второго курса записались на сдачу экзамена.

Всего существует n дней, в которые можно сдавать экзамен. Каждый студент записался для сдачи на один из этих дней. Получилось, что в i-й день экзамен желают сдать a_i студентов, а максимальное количество студентов, у которых преподаватели готовы принять экзамен в этот день, равно b_i.

Преподавателям необходимо обеспечить всем студентам возможность сдавать экзамен, поэтому некоторых студентов может быть придется переместить на другой день. Преподаватели могут выбрать любое множество студентов и назначить каждому из них любой день для сдачи.

Если студент желал сдать экзамен в i-й день, а преподаватели в итоге перенесли его на j-й день, то *недовольство* этого студента будет равно $|i - j|$.

Помогите преподавателям распределить студентов так, чтобы для всех i в i-й день экзамен сдавали не более b_i студентов, а максимальное недовольство среди студентов было минимальным.

Формат входных данных

В первой строке ввода дано одно целое число n — количество дней, в которые можно сдавать экзамен ($1 \leqslant n \leqslant 10^6$).

Во второй строке ввода даны n целых чисел a_i — количество студентов, которые желают сдать экзамен в i-й день ($1 \leqslant a_i \leqslant 10^9$).

В третьей строке ввода даны n целых чисел b_i — максимальное количество студентов, у которых преподаватели готовы принять экзамен в i-й день ($0 \leqslant b_i \leqslant 10^9$).

Формат выходных данных

Выведите единственное целое число — для какого минимального k можно добиться, чтобы недовольство любого студента не превышало k. Если решения не существует, следует вывести -1.

Примеры

стандартный ввод	стандартный вывод
4 6 14 70 1 70 3 16 5	2
1 2 2	0
1 3 2	-1

Замечание

Рассмотрим первый пример. Заметим, что 70 студентов хотят сдавать экзамен в третий день, но суммарно во второй, третий и четвертый день сдать экзамен смогут только 24 студента, поэтому ответ не меньше 2.

Следующий алгоритм перераспределяет студентов по дням, переместив каждого студента не более чем на 2 дня.

- $6, 14, 70, 1$ — исходная запись студентов;
- $6, 70, 14, 1$ — перенесли всех студентов с третьего дня на второй, а всех студентов со второго дня на третий;

- $70, 6, 14, 1$ — перенесли всех студентов со второго дня на первый, а всех студентов с первого дня на второй;
- $70, 6, 11, 4$ — перенесли трех студентов с третьего дня на четвертый;
- $70, 3, 14, 4$ — перенесли трех студентов со второго дня на третий;

Заметим, что каждый студент был перемещен не более чем на два дня от своей исходной записи.

Задача E. Справедливое ограбление

Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Робин Гуд, как известно, крадет у богатых и раздает награбленное бедным. Правда в этот раз, к сожалению, ему придется обойтись только грабежом, потому что в городе, в который он прибыл, живут только богатые люди. Всего в городе живет n богатых людей, их дома расположены вдоль главной улицы города, в i-м доме живет человек с состоянием a_i.

Поскольку Робин Гуд работает вместе с сообщниками, он собирается заранее составить план действий. План состоит из целого числа k и вещественного числа t, которые означают, что будут ограблены дома с номерами $k, k+1, \ldots, n$, и из каждого из них будет украдена доля состояния t. Иными словами, после исполнения такого плана состояния жителей будут равны

$$a^{\text{new}} = [a_1, a_2, \ldots, a_{k-1}, (1-t)a_k, (1-t)a_{k+1}, \ldots, (1-t)a_n],$$

а размер награбленного будет равен

$$b = t \cdot (a_k + a_{k+1} + \ldots + a_n).$$

Назовем *несправедливостью* после ограбления величину $\max(a^{\text{new}}) - \min(a^{\text{new}})$ — разность между максимальным и минимальным состояниями жителей города после ограбления.

Поскольку банда Робина Гуда еще не прибыла целиком в город, он не знает, как много домов они успеют ограбить. Помогите ему для каждого k от 1 до n включительно определить, при каком t от 0 до 1 включительно несправедливость после ограбления по плану (k, t) будет минимальна. Если для заданного k есть несколько значений t, которые минимизируют его несправедливость, выберите то, при котором размер награбленного максимален.

Формат входных данных

В первой строке ввода дано целое число n — количество жителей города ($1 \leqslant n \leqslant 2 \cdot 10^5$).

Во второй строке ввода через пробел перечислены n целых чисел a_i — исходное состояние каждого жителя ($1 \leqslant a_i \leqslant 10^9$).

Формат выходных данных

Выведите через пробел n вещественных чисел t_i ($0 \leqslant t_i \leqslant 1$). Для всех k от 1 до n пара (k, t_k) должна задавать план с минимальной несправедливостью после ограбления среди всех планов с данным k, а среди всех таких — план, при котором в сумме будет украдено как можно больше.

Ваш ответ будет приниматься, если абсолютная или относительная погрешности каждого выведенного числа относительно верного ответа не превосходят 10^{-9}.

Примеры

стандартный ввод	стандартный вывод
3 1 4 2	1.00 0.75 0.50
3 3 2 1	1.00 0.00 0.00

Задача F. Количество антител

Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

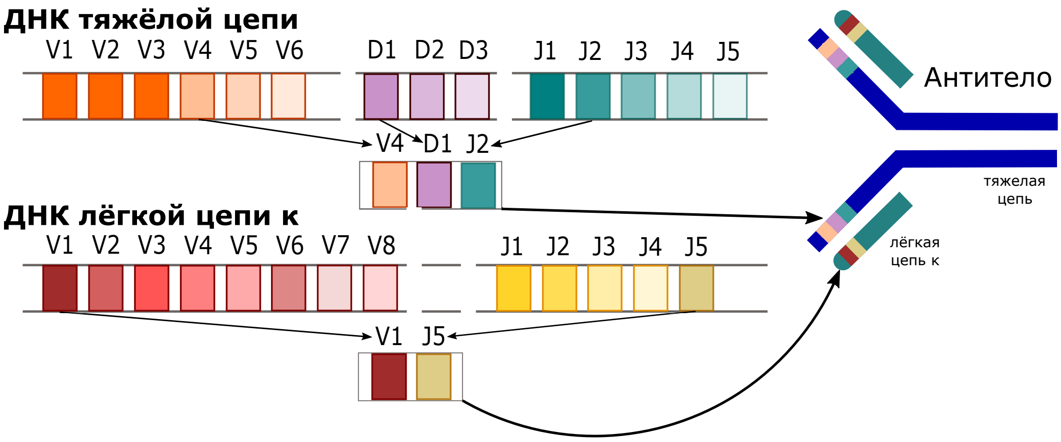
Иммуноглобулины или антитела — белковые молекулы, которые присоединяются к чужеродным агентам в организме и помогают клеткам иммунной системы их обнаружить и ликвидировать. Каждой чужеродной молекуле соответствует свой уникальный иммуноглобулин. Вирусов и бактерий, угрожающих организму, миллионы, поэтому, чтобы никакой враг не остался незамеченным, различных видов антител должно быть очень много, и закодировать каждый иммуноглобулин своей геномной последовательностью в ДНК организма не получается. К счастью в природе нашлось решение.

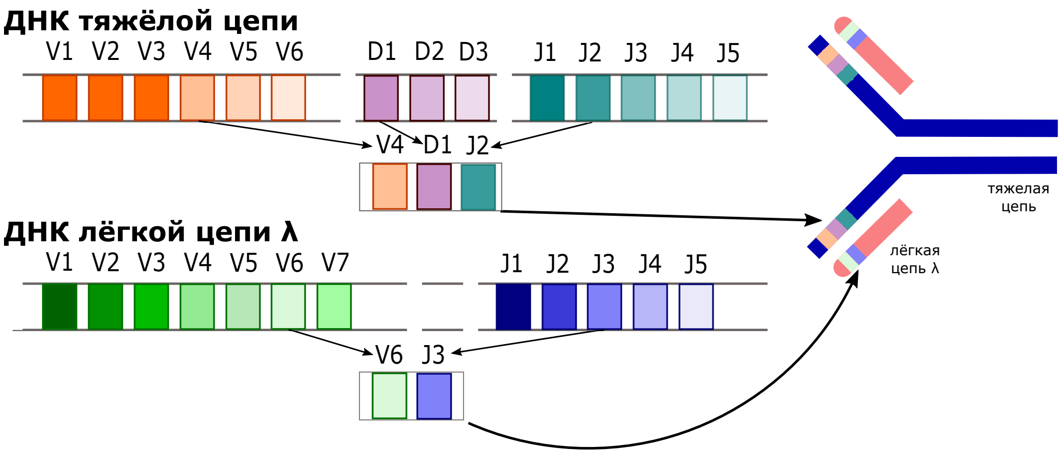
Иммуноглобулины синтезируются клетками иммунной системы В-лимфоцитами на основе генетической информации, которая содержится в ДНК клеток — генах. Чтобы вариантов получилось много, гены, на основе которых синтезируются иммуноглобулины, собираются из нескольких фрагментов как конструктор. Каждый такой фрагмент гена существует в нескольких вариантах и отвечает за свой вариабельный участок молекулы иммуноглобулина. Этот процесс называется соматической рекомбинацией.

Иммуноглобулин состоит из пары одинаковых тяжёлых цепей и пары одинаковых лёгких цепей.

Лёгкая цепь существует двух типов — κ и λ, структура их похожа. Каждый из двух типов лёгкой цепи имеет два вариабельных участка — V и J, для их формирования выбирается один фрагмент из V_κ и J_κ вариантов, соответственно, для лёгкой цепи типа κ, либо из V_λ и J_λ вариантов, соответственно, для лёгкой цепи типа λ.

Существует единственный тип тяжелой цепи, она содержит три вариабельных участка — V, D и J, для формирования каждого из них выбирается один ген из V_h, D_h и J_h вариантов фрагментов, соответственно.





По заданным V_κ, J_κ, V_λ, J_λ, V_h, D_h и J_h требуется определить, сколько вариантов различных иммуноглобулинов может синтезироваться.

Формат входных данных

В первой строке ввода даны целые числа V_κ, J_κ ($1 \leqslant V\kappa, J\kappa \leqslant 1500$) — количество вариантов генных фрагментов для вариабельных участков V и J лёгкой цепи типа κ.

Во второй строке ввода даны целые числа V_λ, J_λ ($1 \leqslant V_\lambda, J_\lambda \leqslant 1500$) — количество вариантов генных фрагментов для вариабельных участков V и J лёгкой цепи λ, соответственно.

В третьей строке ввода даны целые числа V_h, D_h и J_h ($1 \leqslant V_h, D_h, J_h \leqslant 1000$) — количество вариантов генных фрагментов для вариабельных участков V, D и J тяжёлой цепи, соответственно.

Формат выходных данных

Выведите единственное число — количество вариантов иммуноглобулинов, которые могут синтезироваться.

Пример

стандартный ввод	стандартный вывод
40 5 41 6 50 30 6	4014000

Замечание

В заключение отметим, что помимо соматической рекомбинации вариативность иммуноглобулинов обеспечивается и другими механизмами, которые в этой задаче не рассматриваются.

Задача G. Парусная математика

Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Капитан Поликарп с детства мечтает о собственном трехмачтовом парусном корабле. И вот, наконец, его мечта близка к осуществлению — он накопил достаточно денег и приобрел себе прекрасную каракку «Пульхерия». Вот только он не учел, что продавалась она без парусов, и паруса ему теперь предстоит искать отдельно.

Всего на каракке должно быть четыре паруса: по одному на передней и задней мачтах, и два на центральной. А у Поликарпа как раз есть четыре куска ткани площадью t_1, t_2, t_3 и t_4. Поликарп может расположить эти куски ткани на мачтах в любом порядке, возможно, предварительно уменьшив их площадь.

Если обозначить размер паруса на передней мачте за a_1, размеры парусов на центральной мачте — за a_2 и a_3, а размер паруса на задней — за a_4, то *маневренность* корабля определяется как $a_1a_4 + a_2 + a_3$, а *стабильность* — как $a_1 + a_4 + a_2a_3$.

Чтобы каракка могла надежно ходить по морям, капитану необходимо добиться того, чтобы маневренность и стабильность корабля были равны. Ну а для максимального комфорта от использования корабля это значение должно быть как можно больше.

Таким образом, Поликарпу нужно, если необходимо, уменьшить некоторые из значений $t_1, \ldots, t_4$, а затем использовать получившиеся четыре куска ткани в качестве четырех парусов так, чтобы выполнялось равенство $a_1a_4 + a_2 + a_3 = a_1 + a_4 + a_2a_3$, причем это значение должно быть как можно больше.

Помогите Поликарпу найти способ добиться равенства маневренности и стабильности, при этом получившиеся равные значения этих двух величин должны быть максимальными возможными.

Формат входных данных

В единственной строке ввода даны четыре целых числа t_1, t_2, t_3 и t_4 — размеры имеющихся у Поликарпа кусков ткани ($1 \leqslant t_i \leqslant 10^4$).

Формат выходных данных

В первой строке ввода выведите p — перестановку чисел от 1 до 4, i-е число которой равно номеру куска ткани, из которого получается i-й парус. Так, если второй парус был получен из четвертого куска ткани, $p_2 = 4$.

Во второй строке выведите разделенные пробелами числа a_1, a_2, a_3 и a_4 — итоговые размеры парусов ($1 \leqslant a_1, a_2, a_3, a_4 \leqslant 10^4$). Эти числа не обязаны быть целыми.

Ваш ответ будет приниматься, если абсолютная погрешность итоговых значений маневренности и стабильности относительно верного ответа не превосходят $2 \cdot 10^{-6}$ и сами эти значения отличаются друг от друга не более, чем на 10^{-6}.

Примеры

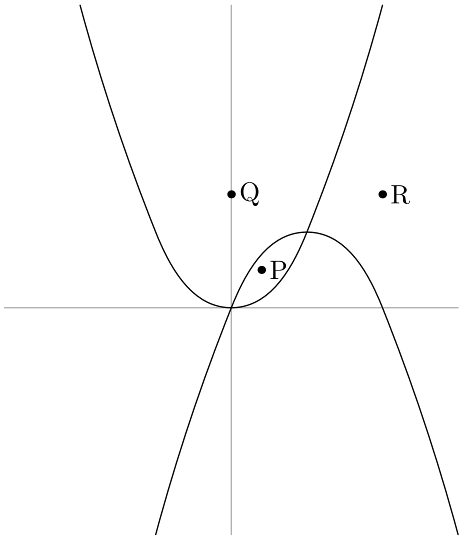
стандартный ввод	стандартный вывод
1 1 1 1	1 2 3 4 1.0 1.0 1.0 1.0
7 5 3 5	1 2 4 3 7.0 4.0 5.0 3.0
2 3 4 5	1 2 3 4 2.0 2.3333333333 4 5

Задача H. Море парабол

Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

На плоскости находится множество парабол,заданных уравнениями вида $y = a \cdot x^2 + b \cdot x + c$.

Будем считать, что точка находится внутри параболы, если при положительном коэффициенте a она находится строго выше параболы, а при отрицательном — строго ниже параболы.



На этом рисунке точка P находится внутри обеих парабол, точка Q внутри одной из двух, а точка R не находится внутри ни одной параболы.

Вам необходимо найти любую точку, которая находится внутри всех парабол. Гарантируется, что такая точка существует.

Формат входных данных

Первая строка содержит одно целое число n ($1 \leqslant n \leqslant 100\,000$) — количество парабол.

В каждой из следующих n строк содержатся по три целых числа a, b, c ($|a|, |b|, |c| \leqslant 10^9$; $a \neq 0$), которые задают параболу $y = a \cdot x^2 + b \cdot x + c$.

Формат выходных данных

Выведите для вещественных числа x и y — координаты точки, которая находится внутри всех парабол.

Ответ считается верным, если существует точка находящаяся на расстоянии не более 10^{-6} от выведенной, такая что она находится строго внутри всех парабол.

Пример

стандартный ввод	стандартный вывод
4 1 2 3 1 -3 -5 -1 3 4 -2 4 6	0.24999999632501932 4.124999990812548

Задача I. Поле чудес

Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Катя давно хотела оказаться на игре «Поле чудес». И вот её мечта исполнилась.
Напомним правила игры «Поле чудес»:

- Ведущий загадывает слово. Игроки по очереди называют букву.
- Если буква есть в слове, то ведущий открывает все вхождения этой буквы в слово и игрок может назвать новую букву.
- Если же игрок назвал букву, которой нет в слове, то ход переходит к следующему игроку.

Ведущий загадал слово длины L. Так как Катя очень хотела победить в игре, она взломала компьютер редактора передачи и выяснила, что загаданное слово будет одним из N слов.

Помогите Кате понять, сможет ли она гарантировано выиграть, если она начинает первой в игре: верно ли, что какое бы не было загадано слово из похищенного Катей списка, она сможет его отгадать, не допустив, чтобы ход перешёл к следующему игроку.

Формат входных данных

В первой строке даны два числа L и N —длина загаданного слова и количество слов, которые выучила Катя ($1 \leqslant L \leqslant 10^6$, $1 \leqslant N \leqslant 10^5$).

В следующих N строках даны различные слова длины L, которые состоят из строчных латинских букв.

Гарантируется, что суммарная длина всех слов не превышает 10^6.

Формат выходных данных

Выведите «`YES`», если Катя может гарантированно победить, иначе выведите «`NO`».

Примеры

стандартный ввод	стандартный вывод
5 2 hello world	YES
4 4 idea play game warm	YES
4 2 game name	NO

Замечание

В первом примере Катя может победить, первой назвав букву «`l`». По открывшимся буквам она однозначно определяет, какое слово загадано.

А вот в третьем примере, какой бы стратегии Катя ни придерживалась, есть риск, что она не угадает очередную букву, и ход перейдет к следующему игроку.

Задача J. Юрик и урок технологии

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Сегодня Юрик с огромной радостью встал с утра пораньше, ведь по расписанию первый урок в школе — технология! Это любимый урок Юрика, на котором он обычно игнорировал учителя и играл с друзьями в настольные игры на задней парте. Но каково было его разочарование, когда выяснилось, что сегодня на уроке состоится контрольная работа.

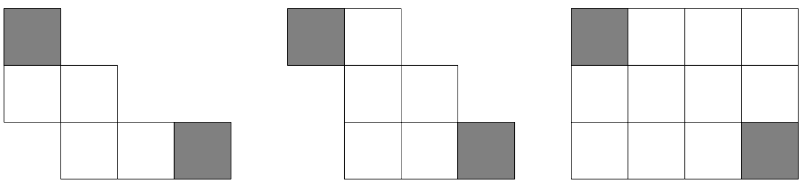
В самом начале урока учитель раздал каждому из школьников, включая Юрика, прямоугольную доску размера $N \times M$. При помощи нарисованных карандашом линий вся доска разделена на N строк и M столбцов. Таким образом, доска состоит из $N \cdot M$ квадратных клеток размера 1×1.

Контрольная работа заключается в том, чтобы вырезать из доски некоторое количество клеток, воспользовавшись лобзиком, так, чтобы оставшаяся часть доски была *красивой*.

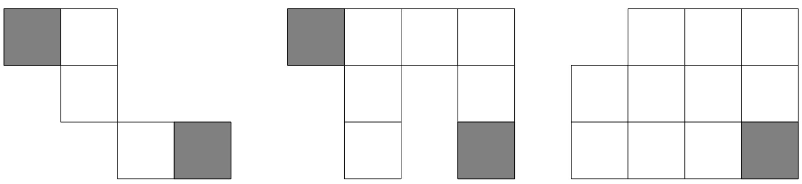
Доска называется *красивой*, если выполнены следующие пять условий:

1. Левая верхняя клетка исходной доски не была вырезана.
2. Правая нижняя клетка исходной доски не была вырезана.
3. Получившаяся доска является связной областью клеток. Это значит, что из любой клетки можно попасть в любую другую клетку за некоторое количество шагов, если за один шаг можно переместиться в соседнюю клетку слева, справа, сверху или снизу.
4. Для любой строки получившейся доски верно, что все клетки, находящиеся в данной строке, которые не были вырезаны, образуют непрерывный отрезок клеток по горизонтали.
5. Для любого столбца получившейся доски верно, что все клетки, находящиеся в данном столбце, которые не были вырезаны, образуют непрерывный отрезок клеток по вертикали.

Любая доска, не удовлетворяющая хотя бы одному из условий, называется *некрасивой*. На изображениях ниже приведены примеры красивых и некрасивых досок. Серым цветом обозначены левая верхняя и правая нижняя клетки.



Примеры красивых досок размера 3×4



Примеры некрасивых досок размера 3×4

Так как Юрик никогда не слушал учителя технологии и увлекался математикой, вместо выполнения данной работы он задумался, сколько различных красивых досок можно получить из исходной доски при помощи вырезания некоторого, возможно, нулевого, количества клеток? Две доски считаются различными, если множества вырезанных из них клеток не совпадают.

Помогите Юрику ответить на этот вопрос.

Формат входных данных

В единственной строке через пробел записаны два целых числа N и M — размеры исходной доски ($1 \leqslant N, M \leqslant 10^5$).

Формат выходных данных

Выведите одно число — количество различных красивых досок, которые Юрик сможет получить из исходной доски при помощи вырезания некоторых клеток.

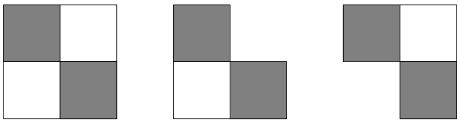
Так как ответ может быть достаточно большим, выведите остаток от деления ответа на число 998 244 353.

Примеры

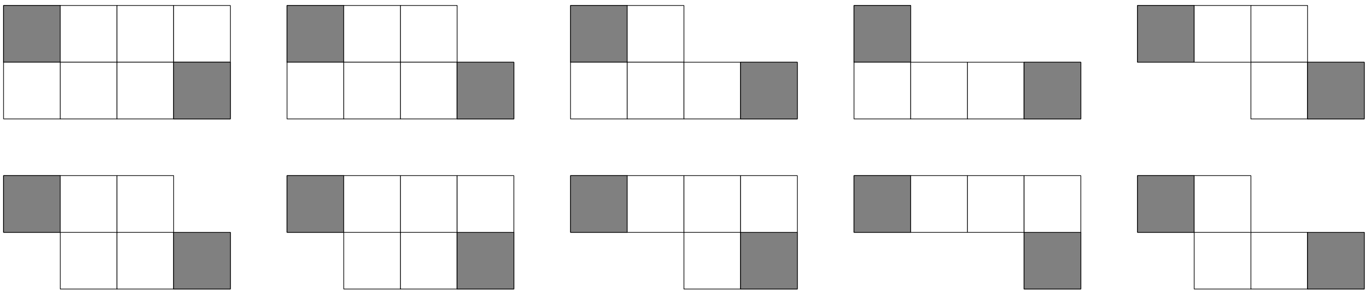
стандартный ввод	стандартный вывод
2 2	3
2 4	10
100 100	818380736

Замечание

Ниже приведены изображения, иллюстрирующие первые два примера из условия:



Все возможные красивые доски, которые можно получить из доски 2×2

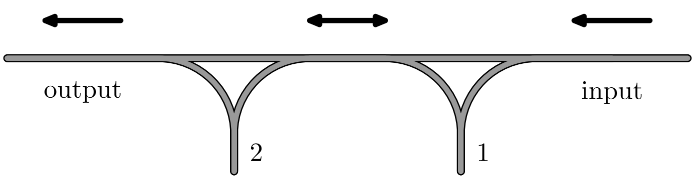


Все возможные красивые доски, которые можно получить из доски 2×4

Задача K. Железнодорожная сортировка

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Арсений работает оператором на сортировочной станции, устройство которой показано на рисунке.



На станции есть входной путь, помеченный на схеме как «`input`», выходной путь, помеченный на схеме как «`output`» и два тупика. Оператор может перемещать вагоны между путями и тупиками, используя следующие операции.

Если вагон x является первым вагоном на входном пути справа от тупиков, этот вагон можно переместить в любой тупик. Команда «`1`» перемещает вагон в тупик 1, а команда «`2`» перемещает вагон в тупик 2.

Если вагон x — ближайший к выезду вагон в одном из тупиков, его можно переместить в выходной путь слева от тупиков. Команда «`-1`» перемещает вагон из тупика 1, а команда «`-2`» перемещает вагон из тупика 2.

Наконец, можно перемещать вагоны между тупиками, если x — ближайший к выезду вагон в одном из тупиков, его можно переместить в другой тупик. Команда «`12`» перемещает вагон из тупика 1 в тупик 2, а команда «`21`» перемещает вагон из тупика 2 в тупик 1.

Обратите внимание, что вагоны не могут быть возвращены в тупик с выходного пути слева от сортировочной станции и не может быть возвращен из тупика на входной путь справа сортировочной станции. Также нельзя проводить вагон напрямую с входного пути на выходной путь, требуется использовать тупик. Оба тупика могут содержать произвольное число вагонов.

Справа на станцию прибывает состав из n вагонов, каждый вагон имеет номер от 1 до n, разные вагоны имеют разные номера.

Арсений должен отсортировать вагоны, чтобы они все находились на выходном пути и их номера слева направо шли по возрастанию. Помогите ему сформировать последовательность команд, которая позволит этого добиться. Количество команд в последовательности не должно превышать $2 \cdot 10^6$.

Формат входных данных

Первая строка ввода содержит число n — количество вагонов ($1 \leqslant n \leqslant 1000$).

Вторая строка содержит n различных целых чисел $a_1, a_2, \ldots, a_n$ ($1 \leqslant a_i \leqslant n$) — номера вагонов в порядке слева направо, в котором находятся на входном пути.

Формат выходных данных

Выведите последовательность команд, которая приведет к тому, что вагоны будут находиться на выходном пути и их номера будут идти по возрастанию. Последовательность должна содержать не более $2 \cdot 10^6$ команд.

Если есть несколько подходящих последовательностей, можно вывести любую из них.

Гарантируется, что для любых входных данных существует последовательность команд, содержащая не более $2 \cdot 10^6$ команд.

Пример

стандартный ввод	стандартный вывод
5 3 1 5 2 4	1 1 -1 1 1 -1 12 -1 1 -1 -2

Задача L. День рождения

Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Богдан получил на день рождения настольную игру «Сумма на подотрезке». Эта увлекательная настольная игра состоит из n двусторонних карточек. На каждой сторонах карточки написано некоторое целое число. Карточки выкладываются в ряд на стол и получают номера с 1 до n слева направо. После этого карточки можно переворачивать, но нельзя менять местами.

Игрок в качестве задания получает номера l и r, а затем каждую карточку между l-й и r-й включительно размещает одной из сторон наверх. Цель игрока — добиться, чтобы сумма чисел на верхних сторонах карточек от l-й и r-й включительно оказалась максимальной.

Богдану быстро наскучило просто искать максимальную сумму, и он решил усложнить игру. Теперь Богдан выбирает число k, при выполнении задания для карточек от l-й и r-й включительно он переворачивает карточки таким образом, что получившаяся сумма чисел была максимальной, но не делилась на k. Если Богдан выполняет задание с карточками от l-й до r-й, то полученную сумму он обозначает как $f(l, r)$. Если от l до r невозможно выбрать числа на карточках так, чтобы полученная сумма была не кратна k, Богдан считает, что $f(l, r) = 0$.

Наигравшись с карточками, Богдан задумался над такой задачей. Он захотел посчитать сумму всех $f(l, r)$ для всех возможных пар l и r, то есть найти $\sum\limits_{1 \leqslant l \leqslant r \leqslant n} f(l, r)$.

Помогите Богдану найти эту сумму. Так как ответ может быть очень большим, вычислите его по модулю $10^9 + 7$.

Формат входных данных

Первая строка входных данных содержит два натуральных числа n и k ($1 \leqslant n \leqslant 5 \cdot 10^5$; $1 \leqslant k \leqslant 10^9$).

Следующие n строк содержат описание карточек на столе: каждая строка содержит два натуральных числа a_i и b_i ($1 \leqslant a_i, b_i \leqslant 10^9$) — числа на сторонах карточки с номером i.

Формат выходных данных

Выведите одно число — ответ на задачу по модулю $10^9 + 7$.

Примеры

стандартный ввод	стандартный вывод
3 3 1 2 2 3 3 1	23
5 5 4 1 4 2 2 3 2 4 1 5	130