Задача A. Камни и бананчики

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	512 мегабайт

У мальчика Пети есть клетчатый листок бумаги размером $n \times n$, столбцы и строки листа пронумерованы числами от 1 до n, слева-направо и сверху-вниз, соответственно. Таким образом, каждая клетка задается парой чисел (x, y), где x задает номер строки, в которой лежит эта клетка, а y — номер столбца.

Петя положил два бананчика в клетки (x_1, y_1) и (x_2, y_2), после чего на некоторых клетках листа магическим образом появились раскалённые камни. Петя может успеть переложить не более k камней до того, как они прожгут бумагу.

Каждый камень прожигает ту клетку листа, где он находится **после перекладывания**, из-за чего лист может развалиться на несколько кусков. Петя не хочет, чтобы бананчики грустили, поэтому они должны в итоге оказаться на одном куске бумаги. Помогите Пете переложить камни так, чтобы бананчики не грустили, или скажите, что это невозможно.

Формат входных данных

В первой строке содержатся два целых числа n и k ($1 \leqslant n \leqslant 1000$, $0 \leqslant k \leqslant 5$).

Во второй строке содержатся два числа x_1 и y_1 ($1 \leqslant x_1, y_1 \leqslant n$) — координаты первого бананчика.

В третьей строке содержатся два числа x_2 и y_2 ($1 \leqslant x_2, y_2 \leqslant n$) — координаты второго бананчика.

Далее следуют n строк, в i-й из которых находится бинарная строка длины n, j-й символ этой строки равен 1, если в клетке (i, j) лежит камень, и равен 0 в ином случае.

Гарантируется, что в клетках (x_1, y_1) и (x_2, y_2) нет камней.

Формат выходных данных

В первой строке выведите «`Yes`», если Петя может добиться того, чтобы бананчики не грустили, и «`No`» в ином случае.

Если вы вывели «`Yes`», то во второй строке выведите число $m (0 \leqslant m \leqslant k)$ — количество камней, которые надо передвинуть Пете.

Дальше выведите m строк, в каждой по четыре числа f_x, f_y, p_x, p_y ($1 \leqslant f_x, f_y, p_x, p_y \leqslant n$), которые означают, что Пете следует переместить камень из клетки (f_x, f_y) в клетку (p_x, p_y).

Обратите внимание, что нельзя перекладывать один и тот же камень дважды, а так же перекладывать камень в клетку с бананом или в клетку, которая уже содержит другой камень или содержала его ранее.

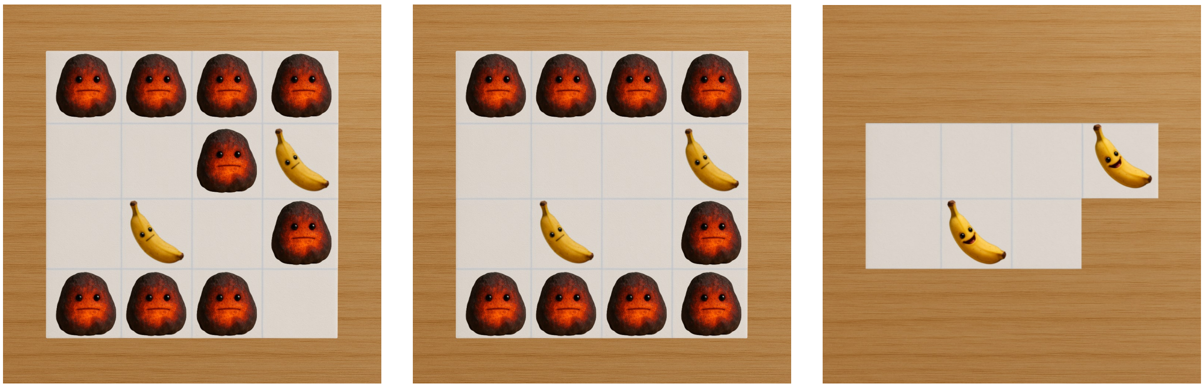
Если существует несколько подходящих ответов, то выведите любой из них.

Примеры

стандартный ввод	стандартный вывод
4 1 3 1 2 4 1111 0010 0001 1110	Yes 1 2 3 4 4
2 0 1 1 2 2 01 10	No

Замечание

В первом примере Пете следует переместить камень с позиции $(2, 3)$ на позицию $(4, 4)$.



На рисунках изображены: изначальная расстановка бананчиков и камней, расстановка после перемещения и уцелевший кусок листа с двумя довольными бананчиками

Во втором примере Петя не может перемещать камни, поэтому лист распадется на два куска, в каждом из которых лежит по бананчику. Клетки $(1, 1)$ и $(2, 2)$ не образуют связную фигуру, так как являются соседними по углу, а не по стороне.

Задача B. Кодовый замок

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Ульяна очень беспокоится за сохранность своих вещей, поэтому пользуется кодовыми замками.

Кодовый замок представляет собой набор из n роторов, на i-м из которых написаны числа от 0 до $m_i - 1$ включительно. Состояние замка определяется массивом чисел $a_1, \ldots, a_n$ $(0 \leqslant a_i < m_i)$ — числа, установленные на каждом из роторов. Изначально все a_i равны 0.

Ульяна установила комбинацию $b_1, \ldots, b_n$ $(0 \leqslant b_i \leqslant m_i - 1)$ в качестве кода замка. Замок откроется, если в результате манипуляций с замком окажется, что $a_i = b_i$ для всех $i \in \{1, 2, \ldots, n\}$.

За одно действие Ульяна может выбрать отрезок $[l, r]$ $(1 \leqslant l \leqslant r \leqslant n)$ и *прокрутить* все роторы на этом отрезке, после чего для каждого i $(l \leqslant i \leqslant r)$ значение a_i заменится на $(a_i + 1) \bmod m_i$.

Ульяна хочет узнать, насколько быстро она сможет открыть свой замок, в случае необходимости. Помогите ей узнать, какое минимальное количество действий ей для этого потребуется.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. Первая строка содержит единственное целое число t $(1 \leqslant t \leqslant 500)$ — количество наборов входных данных. Далее следуют их описания.

В первой строке каждого набора содержится число n $(1 \leqslant n \leqslant 500)$ — количество роторов замка.

Во второй строке каждого набора содержатся числа $m_1, m_2, \ldots, m_n$ $(2 \leqslant m_i \leqslant 500)$ — количество возможных значений в каждом из роторов.

В третьей строке каждого набора содержатся числа $b_1, b_2, \ldots, b_n$ $(0 \leqslant b_i \leqslant m_i - 1)$ — комбинация значений, которая разблокирует замок.

Гарантируется, что сумма n по всем наборам входных данных не превосходит 500.

Формат выходных данных

Для каждого замка выведите в отдельной строке минимальное количество действий, которое требуется для его открытия.

Пример

стандартный ввод	стандартный вывод
3 4 5 5 5 5 1 2 3 4 5 10 2 10 2 10 8 1 8 1 8 7 4 4 3 2 3 4 4 1 3 0 1 0 3 1	4 10 3

Замечание

Обозначим за $[l, r]$ действие на отрезке с l по r.

Первый замок можно открыть за 4 последовательных действия: $[1, 4], [2, 4], [3, 4], [4, 4]$.

Второй замок можно открыть, выполнив 8 действий $[1, 5]$, 1 действие $[2, 2]$ и 1 действие $[4, 4]$.

Третий замок можно открыть, выполнив 3 последовательных действия: $[1, 7], [2, 6], [2, 6]$.

Задача C. Хорошие раскраски – 7

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Клетки таблицы $n \times n$ раскрашены в три цвета: красный, синий и белый. Требуется перекрасить клетки белого цвета в синий или красный цвет (разные клетки можно перекрашивать в разные цвета) так, чтобы выполнялись следующие условия:

- В любой строке находится **четное** количество красных клеток;
- В любом столбце находится **четное** количество красных клеток;
- На главной диагонали таблицы находится **четное** количество красных клеток;
- На побочной диагонали таблицы находится **четное** количество красных клеток.

Главной и побочной диагональю квадрата $n \times n$ называется пара диагоналей, каждая из которых содержит ровно по n клеток.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится одно целое число t ($1 \leqslant t \leqslant 2\,000$) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит число n ($1 \leqslant n \leqslant 1\,000$) — размер таблицы.

Следующие n строк содержат по n записанных подряд символов «`R`», «`B`», «`W`» — цвета клеток в таблице:

- Символ «`R`» обозначает, что соответствующая клетка красная;
- Символ «`B`» обозначает, что соответствующая клетка синяя;
- Символ «`W`» обозначает, что соответствующая клетка белая.

Гарантируется, что сумма n по всем наборам не превосходит 10^4.

Формат выходных данных

Выведите «`Yes`» (без кавычек), если можно перекрасить белые клетки таблицы так, чтобы выполнялось условие задачи и «`No`» в противном случае.

В случае положительного ответа на задачу выведите n строк по n символов «`R`» и «`B`» — искомую раскраску таблицы.

Пример

стандартный ввод	стандартный вывод
3 3 RBR WWW WWR 2 BW WR 4 WRWB WWWW RWWW BBWW	Yes RBR BBB RBR No Yes RRBB BRBR RBRB BBRR

Задача D. Как обидеть черепашку

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

Дана таблица $h \times w$, строки которой пронумерованы сверху-вниз числами от 1 до h, а столбцы пронумерованы числами от 1 до w слева-направо. Будем обозначать парой (i, j) клетку, которая лежит на пересечении i-й строки и j-го столбца.

Черепашка живет в клетке $(1, 1)$ и хочет добраться до клетки (h, w), где живет ее подружка. Как известно, черепашки, живущие в таблице, могут передвигаться только слева-направо и сверху-вниз. А именно, из клетки (i, j) черепашка может попасть в клетки $(i+1, j)$ и $(i, j+1)$, если, конечно, эти клетки существуют.

Злой Саша решил заблокировать некоторые клетки таблицы так, чтобы черепашка **не смогла** добраться до своей подружки. Черепашка, как можно догадаться, не сможет посещать заблокированные клетки. Но Саша не хочет показаться злодеем, а поэтому не будет блокировать никакие две соседние по стороне или углу клетки, а также не будет блокировать клетки $(1, 1)$ и (h, w).

Проверьте, сможет ли Саша заблокировать некоторые клетки так, чтобы черепашка не смогла добраться из клетки $(1, 1)$ в клетку (h, w).

Формат входных данных

В единственной строке входных данных указана пара чисел h и w ($2 \leqslant h, w \leqslant 200$).

Формат выходных данных

Выведите «`No`», если Саша не сможет заблокировать путь черепашке.

В ином случае выведите «`Yes`» в первой строке, а в следующих h строках выведите по w символов «`X`» или «`.`» — таблицу с отмеченными заблокированными клетками. Символ «`.`» означает обычную клетку, а символ «`X`» — заблокированную.

Примеры

стандартный ввод	стандартный вывод
`2 2`	`No`

`3 7`	`Yes`
	`.X...X.`
	`...X...`
	`.X...X.`

Задача E. Бойцовский клуб

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Как известно, гориллы — сильные и умные животные. После того, как ученые научили одну из горилл языку программирования Zig, она преисполнилась в своем сознании, выбралась на волю, и объединила всех горилл мира в обществе будущего. Ясно, что свои знания надо передавать следующим поколениям, поэтому горилла основала университет с единственным факультетом, где она преподавала язык Zig и алгоритмы в распределенных системах.

Всего в университет захотели поступить n горилл, силы горилл равны, соответственно, $a_1, \ldots, a_n$. Считается, что нет двух равных по силе горилл, поэтому a_i — различные числа от 1 до n.

Поступление организовано следующим образом. Сначала выбирается произвольный отрезок $[l, r]$ ($l < r$), на котором объявляется, что две самые сильные гориллы поступают в университет. Гориллы не только сильные и умные животные, но еще и самые целеустремленные, поэтому, если хотя бы одна из двух самых сильных горилл находится не на краю отрезка, то ей тут же дают отпор, и она выбывает из конкурса.

Горилла-основатель понимает это, а поэтому старается выбирать для конкурса только *честные* отрезки, то есть те, в которых два наибольших значения встречаются на границах отрезка. Иными словами, отрезок $[l, r]$ *честный*, если $\min\{a_l, a_r\} > a_i$ для всех $l < i < r$. В частности отрезок из двух горилл всегда является *честным*.

Но пока горилла-основатель думает, как устроить конкурс, остальные гориллы непрерывно тренируются. В итоге это приводит к тому, что в последующие q часов происходит следующее: в j-й час ($1 \leqslant j \leqslant q$) горилла с номером x_j идет в спортивный зал и становится новой самой сильной гориллой, после чего в этот и все последующие часы (до того, как она снова пойдет в спортивный зал) ее сила полагается равной $n + j$.

Теперь горилла-основатель совершенно запуталась и хочет хотя бы понять, сколько способов у нее будет провести конкурс *честно* в каждый из моментов времени.

Иными словами для каждого момента времени j ($1 \leqslant j \leqslant q$) требуется вычислить количество *честных* отрезков в момент времени j.

Формат входных данных

В первой строке заданы два целых положительных числа n и q ($1 \leqslant n, q \leqslant 10^6$) — количество горилл-абитуриентов и количество часов.

Во второй строке находятся n целых положительных чисел $a_1, \ldots, a_n$ ($1 \leqslant a_i \leqslant n$) — силы горилл в начальный момент времени. Гарантируется, что все числа a_i различны.

В третьей строке находятся q чисел $x_1, \ldots, x_q$ ($1 \leqslant x_i \leqslant n$), где x_j — номер гориллы, которая идет в спортивный зал в j-й час.

Формат выходных данных

В единственной строке выведите q чисел, i-е из которых равно количеству *честных* отрезков в момент времени i.

Пример

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

Задача F. Арсен и солдатики

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Арсен командует отрядом из n игрушечных солдат. Каждое утро и каждый вечер проводится построение. На каждом построении каждому солдату выдают ружье одного из n видов.

Перед началом дня у Арсена есть два частично заполненных расписания: $a_1, \ldots, a_n$ и $b_1, \ldots, b_n$. Здесь a_i — вид ружья, который получает i-й солдат на утреннем построении, а b_i — вид ружья, который он получает на вечернем построении. Если a_i или b_i равно -1, это означает, что вид ружья на соответствующее построение ещё не назначен. Арсен может заменить каждое -1 любым числом от 1 до n.

После того, как расписания выдачи ружей полностью определены, Арсен проводит еще один ритуал: он решает для каждого солдата, выдать ему мыло или нет. Формально, он создает бинарный массив c, где $c_i = 0$ или $c_i = 1$. Этот массив должен удовлетворять двум условиям баланса:

- Для каждого вида ружья k (от 1 до n), среди всех солдат, которые получили ружье вида k на утреннем построении, количество солдат с мылом ($c_i = 1$) должно совпадать с количеством солдат без мыла ($c_i = 0$).
- Аналогично, для каждого вида ружья k, среди всех солдат, которые получили ружье вида k на вечернем построении, также должно быть поровну солдат с мылом и без.

Помогите Арсену определить, можно ли так дозаполнить расписания a и b (заменив все -1 на числа от 1 до n), чтобы существовал хотя бы один способ выдачи мыла (массив с), удовлетворяющий условиям баланса. Если это возможно, восстановите пример подходящих массивов a, b и c.

Формат входных данных

Первая строка содержит целое число n ($1 \leqslant n \leqslant 10^6$) — количество солдат.

Вторая строка содержит n целых чисел $a_1, a_2, \ldots, a_n$ ($1 \leqslant a_i \leqslant n$ или $a_i = -1$) — назначения ружей для утреннего построения.

Третья строка содержит n целых чисел $b_1, b_2, \ldots, b_n$ ($1 \leqslant b_i \leqslant n$ или $b_i = -1$) — назначения ружей для вечернего построения.

Формат выходных данных

В первой строке выведите «Yes», если можно дозаполнить массивы a и b, чтобы пара массивов стала хорошей. В противном случае выведите «No».

Если ответ «Yes», выведите три дополнительные строки:

- Во второй строке выведите n целых чисел — полный массив a после заполнения всех -1.
- В третьей строке выведите n целых чисел — полный массив b после заполнения всех -1.
- В четвертой строке выведите n целых чисел (0 или 1) — массив c, удовлетворяющий условиям баланса.

Если существует несколько возможных ответов, выведите любой из них.

Примеры

стандартный ввод	стандартный вывод
6 -1 1 -1 2 -1 2 1 -1 2 -1 2 -1	YES 1 1 1 2 1 2 1 1 2 1 2 1 0 1 0 1 1 0
4 1 1 1 1 1 1 1 2	NO

Задача G. Геометрия!

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Антон не любит длинные легенды, зато любит геометрию, поэтому вот формальное условие задачи.

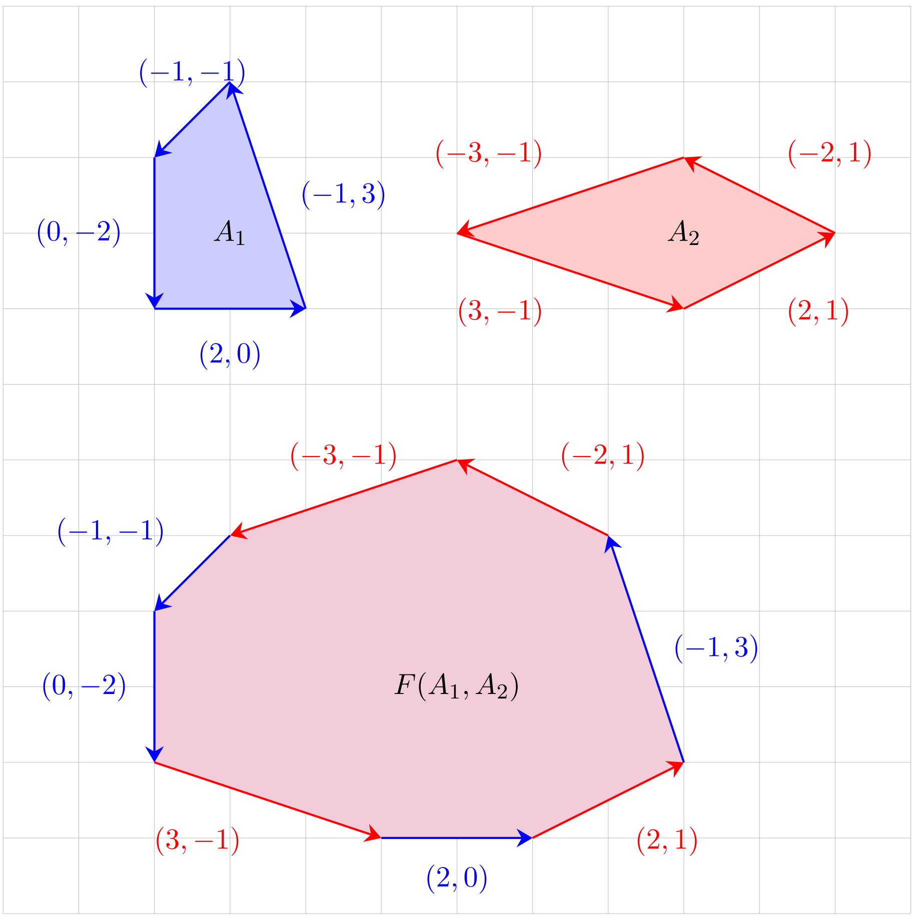
Вам даны n выпуклых многоугольников A_1, A_2, $\ldots$, A_n. Многоугольник номер i состоит из k_i вершин, и, соответственно, k_i ребер. При этом **гарантируется, что никакие две стороны многоугольников не параллельны** (т. е. ни у какого многоугольника нет параллельных сторон, и стороны разных многоугольников также не параллельны). Каждый многоугольник задается как набор сторон, каждая сторона задается как вектор, векторы перечислены в порядке обхода против часовой стрелки (соответственно направления векторов тоже идут против часовой стрелки).

Зададим функцию $F(A, B)$, где A, B — это выпуклые многоугольники. $F(A, B)$ — это выпуклый многоугольник, который составлен из всех сторон многоугольников A и B, то есть в нем каждая сторона — это одна из сторон из A или B, при этом направление и длина этой стороны должны быть сохранены, и каждая сторона должна быть использована ровно один раз. Можно доказать, что всегда существует ровно один такой выпуклый многоугольник.

Зададим функцию $G(A, B)$, где A, B — это выпуклые многоугольники. $G(A, B)$ — это количество таких вершин в $F(A, B)$, для которых одна из смежных сторон взята из A, а другая из B (заметим, что так как нет параллельных сторон, то всегда можно определить, из какого многоугольника взята та или иная сторона).

Вам нужно посчитать $\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} G(A_i, A_j)$

Для лучшего понимания рассмотрим картинку из первого примера.



Здесь нарисованы многоугольники A_1 и A_2, а также $F(A_1, A_2)$. Легко видеть, что в восьмиугольнике $F(A_1, A_2)$ есть ровно 6 вершин, которые смежны со сторонами, взятыми из разных многоугольников. Значит $G(A_1, A_2) = 6$.

Формат входных данных

В первой строке вводится целое число T — количество независимых тестовых случаев ($1 \leqslant T \leqslant 10^5$). Далее идут описания T тестовых случаев.

Описание каждого тестового случая начинается с числа n в отдельной сторке — количества многоугольников ($2 \leqslant n \leqslant 10^5$).

Далее идут описания многоугольников. Для каждого многоугольника в первой сторчке вводится число k_i — количество вершин в многоугольнике. Далее следуют k_i ($3 \leqslant k_i \leqslant 10^6$) строчек, в каждой из которых описывается очередное ребро многоугольника. В каждой строчке через пробел записана пара чисел x и y ($-10^9 \leqslant x, y \leqslant 10^9$) — координаты вектора, задающего очередное ребро. Гарантируется, что эти векторы задают корректные выпуклые многоугольники.

Гарантируется, что сумма n во всех тестовых случаях не превосходит 10^5. Гарантируется, что сумма всех k_i во всех тестовых случаях не превосходит 10^6.

Формат выходных данных

Для каждого тестового случая выведите в отдельной строке одно целое число — $\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} G(A_i, A_j)$.

Пример

стандартный ввод	стандартный вывод
1 2 4 -1 -1 0 -2 2 0 -1 3 4 2 1 -2 1 -3 -1 3 -1	6

Задача H. Трамвайная система

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Мальчик Миша после прочтения Мастера и Маргариты стал бояться трамваев, рельсов и всего с этим связанного. Чтобы преодолеть свой страх, Миша решил построить трамвайную систему в известной компьютерной игре про кубики и изучить свойства построенной системы.

Миша выбрал четное число n и построил n^2 станций, формирующих сетку $n \times n$, затем соединил их с помощью $\frac{n}{2}$ зацикленных не пересекающихся рельсовых путей, образующих вложенные квадраты.

На каждой из n^2 станций Миша разместил по одному трамваю, каждый из которых имел один из 26 типов, обозначаемых строчными буквами латинского алфавита. Ввиду того, что станции образуют сетку $n \times n$, легко записать в виде таблицы типы всех расставленных трамваев.

Стоит уточнить, что Миша боялся не столько самих трамваев, сколько *темных сил*. Поэтому он внимательно следил за величиной *мистики*, которую можно вычислить следующим образом:

Рассматривается таблица с записанными типами трамваев у каждой из станций, затем все типы трамваев выписываются в одну строку s (сначала выписывается первая строка таблицы, затем вторая и так далее). Уже при вычисленной строке s *мистика* определяется как длина наибольшего префикса s, который совпадает с развернутым суффиксом s такой же длины.

Миша понимает, что если запустить трамвайную систему, то трамваи поменяют свое положение, а именно, на каждом из путей трамваи могут сместиться по циклу, при чем на разных путях сдвиг может произойти на разную величину. По всем возможным положениям трамваев Мишу интересует наибольшее возможное значение *мистики*.

К сожалению, Миша не такой умный, как вы, а поэтому просит вас ему помочь и вычислить интересующее его значение.

Формат входных данных

В первой строке указано **четное** натуральное число n ($2 \leqslant n \leqslant 1000$).

В следующих n строках указана конфигурация трамвайной системы — очередная строка содержит n строчных латинских символов, записанных подряд — типы трамваев, которые располагаются в этой строке.

Формат выходных данных

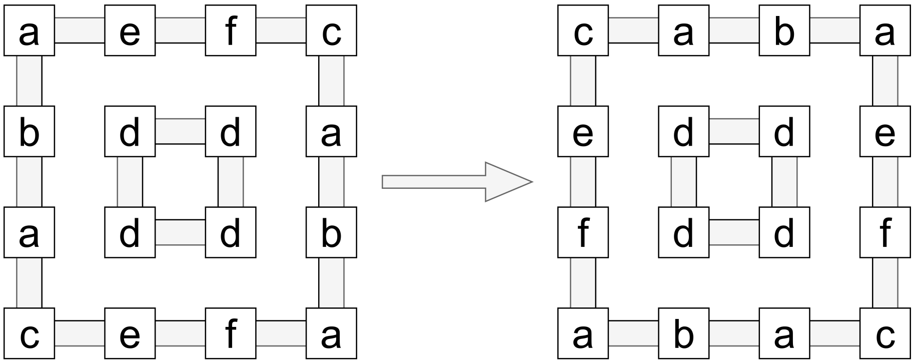
В единственной строке выведите единственное число — наибольшее возможное значение *мистики*.

Примеры

стандартный ввод	стандартный вывод
4 aefc bdda addb cefa	4
4 aaaa aaaa aaaa aaaa	16

Замечание

Рассмотрим первый пример. В нем можно циклически сдвинуть внешнюю трамвайную линию на 3 поезда по часовой стрелке. Тогда получится следующая ситуация:



Если после этого выписать все типы поездов в одну строку, то получится строка `cabaedbefddfabac`. Видно, что в этой строке префикс длины 4 совпадает с развернутым суффиксом длины 4, поэтому *мистика* после указанных манипуляций равна 4. Можно показать, что большее значение *мистики* получить невозможно.

Во втором примере можно циклически сдвигать трамваи произвольным образом, в любом случае выписанная строка будет иметь вид `aaa ...aaa`, она сама является своим суффиксом и префиксом, а также является палиндромом, поэтому *мистика* равна 16.

Задача I. Оценки Рика

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

Морти скатился и начал получать плохие оценки в школе. Рик в первый раз в жизни не забил на это и решил помочь. Для мотивации Морти он хочет доказать, что сам Рик учился лучше.

В школьном журнале за много лет не произошло изменений, так что и Рик, и Морти получали оценки от 1 до 5, либо 0 — аналог российской «Н», означающий, что ученик отсутствовал на уроке.

Известно, что за последние n дней Морти получил n оценок (по одной в день), которые представляют собой массив $a_1, a_2, \ldots, a_n$.

Рик хочет доказать Морти, что он учился лучше. В своё время за те же n дней Рик тоже получил n оценок, которые представляют собой массив $b_1, b_2, \ldots, b_n$ из чисел от 0 до 5.

Рик очень хитрый, поэтому хочет переставить местами свои оценки так, чтобы в каждый из дней его оценка была **не меньше** оценки Морти. При этом, если после перестановки у Рика или Морти в журнале за какой-то день стоит 0, то в этот день Рику **не важно**, кто и какую оценку получил.

Более формально, Рик хочет найти массив c, который получается путем перестановки элементов массива b, такой, что для каждого индекса i выполнено **хотя бы одно** из трёх условий: $c_i \geqslant a_i$, $a_i = 0$ или $c_i = 0$.

Помогите Рику найти подходящий порядок элементов из b или скажите, что это невозможно.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится одно целое число t ($1 \leqslant t \leqslant 10^5$) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит целое число n ($1 \leqslant n \leqslant 3 \cdot 10^5$).

Вторая строка содержит n целых чисел $a_1, a_2, \ldots, a_n$ ($0 \leqslant a_i \leqslant 5$) — оценки Морти.

Третья строка содержит n целых чисел $b_1, b_2, \ldots, b_n$ ($0 \leqslant b_i \leqslant 5$) — оценки Рика.

Гарантируется, что сумма n по всем наборам входных данных не превосходит $3 \cdot 10^5$.

Формат выходных данных

Для каждого набора входных данных в единственной строке выведите элементы массива b в **любом подходящем** порядке.

Если Рик не сможет подобрать нужную ему перестановку оценок, выведите -1.

Примеры

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

Задача J. Упражнение для Дани

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Очередным вечером понедельника Даня сидел в библиотеке и решал задачи по программированию. Он очень любил проводить время за решением задач, но больше он любил гордиться уже решенными задачами перед другими ребятами. Его знакомый знал, что Даня мастер в своем деле и попросил его помочь с одним незамысловатым упражнением.

Даня впал в ступор от таких *упражнений*, но терять репутацию опытного программиста ему не хочется. Поэтому он обратился к вам за помощью. А задача была такая:

Рассмотрим *функцию минимального нетривиального делителя* $M(x) = d$, где d— минимальное число больше единицы, на которое делится x. У единицы нет таких делителей, поэтому полагается $M(1) = 0$.

Например, $M(12) = 2$, $M(35) = 5$, $M(179) = 179$.

По заданному массиву $a_1, \ldots, a_n$ длины n требуется для каждого k от 1 до n вычислить значение следующего выражения:

$$\sum_{i=1}^{k-1} \sum_{j=i+1}^{k} M\left(\frac{a_i a_j}{\gcd(a_i, a_j)^2}\right)$$

Здесь $\gcd(x, y)$ обозначает *наибольший общий делитель (НОД)* чисел x и y.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. Первая строка содержит одно целое число t ($1 \leqslant t \leqslant 10^4$) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит одно целое число n ($1 \leqslant n \leqslant 150\,000$) — размер массива.

Следующая строка каждого набора входных данных содержит n чисел a_i ($2 \leqslant a_i \leqslant 1\,000\,000$) — сам массив a.

Гарантиуретcя, что сумма n по всем наборам не превосходит $150\,000$.

Формат выходных данных

Для каждого набора входных данных выведите n чисел — значение искомый суммы для каждого $1 \leqslant k \leqslant n$.

Пример

стандартный ввод	стандартный вывод
4 3 2 4 8 5 2 3 4 5 6 10 10 12 15 20 10 13 17 11 25 30 5 999979 999749 999959 999917 999671	0 2 6 0 2 6 13 22 0 2 6 13 19 30 54 87 113 133 0 999749 2999457 5999040 9997724

Задача K. Белка и ступеньки

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Злой хозяин Саша выгнал свою кошку Белку на улицу. Теперь Белке нужно обратно попасть домой. Саша живет на втором этаже своего дворца. Чтобы к нему попасть, нужно пройти через n ступенек. Изначально Белка стоит на ступеньке с номером 0, и ей нужно попасть на ступеньку номер n. Белка не очень длинная, поэтому, расстояние между любыми двумя ее лапками должно быть не больше 2. За один шаг кошка переносит одну из своих лапок на следующую ступеньку.

Более формально это значит, что в каждый момент для любых позиций ее лапок l_i и l_j, $|l_i - l_j| \leqslant 2$.

Как у любой кошки, у Белки разные лапки: левая передняя, правая передняя, левая задняя, правая задняя, и они различимы. Вам нужно помочь Белке посчитать количество способов подняться на ступеньку n. Два способа являются различными, если на каком-то шаге Белка переставила разные лапки. Изначально все лапки находятся на ступеньке с номером 0.

Так как количество способов может быть достаточно большим, выведите его по модулю $10^9 + 7$.

Формат входных данных

В первой строке дано одно целое число t ($1 \leqslant t \leqslant 1\,000$) — количество тестовых наборов.

В единственной строке каждого набора данных дано одно целое число n ($1 \leqslant n \leqslant 1\,000\,000$) — количество ступенек, через которые нужно пройти Белке.

Обратите внимание, что нет дополнительного ограничения на сумму n по всем наборам входных данных.

Формат выходных данных

Для каждого набора выведите одно число — количество вариантов добраться до ступеньки n по модулю $10^9 + 7$.

Пример

стандартный ввод	стандартный вывод
3 1 2 6	24 2520 151698586

Задача L. Векторная магия

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

Все знают про МОП — Международное Общество Приматов. Один из членов этого общества, макака Кирилл, занимается изучением магии.

Сегодня, подрабатывая уборщиком в стойле, он обнаружил странный листок, на котором было написано n целых чисел $a_1, \ldots, a_n$. Он сразу понял, что это магическое заклинание! Чтобы его использовать, надо возле каждого числа положить по волшебной руне и щелкнуть пальцами. После этого листок исчезнет и появятся $\frac{(a,b)}{\|b\|_2}$ бананов, где a — вектор в n-мерном пространстве, состоящий из чисел с листка, b — вектор в том же пространстве, состоящий из мощностей волшебных рун, лежащих возле соответствующих чисел на листке.

Кирилл уже много лет изучает магию, а поэтому он уже давно знает значения этих формул, но математику он не изучал, а поэтому просит вас ему помочь. Вам нужно вычислить значение следующего выражения:

$$\frac{(a,b)}{\|b\|_2} = \frac{a_1 \cdot b_1 + a_2 \cdot b_2 + \ldots a_n \cdot b_n}{\sqrt{b_1^2 + b_2^2 + \ldots + b_n^2}}$$

Кирилл тоже не знает, что все это значит, просто это магия

К сожалению, у Кирилла есть только руны мощностей $-3, -1, 1, 3$, зато каждой по n, то есть он может составить любой вектор b, состоящий из этих чисел. Помогите Кириллу разложить руны оптимально, чтобы получить как можно больше бананов!

Формат входных данных

В первой строке дано одно целое число n $(1 \leqslant n \leqslant 10^4)$ — количество чисел на листке.

Во второй строке даны n целых чисел a_i $(-300 \leqslant a_i \leqslant 300)$ — числа, написанные на листке.

Формат выходных данных

Выведите оптимальный вектор (лучший выбор рун) $b_1, \ldots b_n$, где $b_i \in \{-3, -1, 1, 3\}$.

Если оптимальных векторов несколько, то вы можете вывести любой из них.

Примеры

стандартный ввод	стандартный вывод
`3` `1 1 -1`	`1 1 -1`
`3` `-3 20 1`	`-1 3 1`
`5` `0 0 0 0 0`	`-1 -1 -1 -1 -1`