

Problem A. Virus

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 256 mebibytes

Иммунная система губки Боба представляет собой прямоугольник $n \times m$ из иммунных клеток. Губка Боб не хочет идти на учебу завтра и для этого ему нужно заболеть.

Для этого он готов заразить некоторые из своих иммунных клеток вирусом ОРВИ. После заражения происходит процесс распространения вируса: каждую миллисекунду все здоровые клетки, имеющие двух или более заражённых соседей, заражаются, а с заражёнными клетками больше ничего не происходит. Здесь мы называем клетки соседними, если они имеют общую сторону. Сам губка Боб заболевает, только если все его иммунные клетки будут заражены. Поскольку заражать свои иммунные клетки вирусом больно, губка Боб хочет минимизировать количество изначально зараженных вирусом клеток.

Найдите и выведите какой-нибудь способ начального заражения с минимальным возможным числом заражённых клеток, приводящего к болезни губки Боба.

Input

В единственной строке даны два целых числа n и m ($1 \leq n, m \leq 10^3$) — размеры иммунной системы губки Боба.

Output

Выведите описание любого начального заражения с минимальным числом зараженных клеток, которое приводит к болезни губки Боба. В n строках выведите по m символов: 1, если соответствующая клетка иммунной системы изначально заражена, и 0, если нет. Не разделяйте символы пробелами или другими разделителями.

Examples

standard input	standard output
1 3	101
2 1	1 1

Problem D. Long Nim

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 256 mebibytes

Два игрока играют в известную игру Ним. У них есть n кучек камней, в i -ой из них a_i камней. Игроки ходят по очереди, каждый ход соответствующий игрок забирает некоторое положительное количество камней из любой одной кучки. Цель игрока — забрать последний камень. Очевидно, что результат игры предопределен, если оба игрока действуют оптимально. В данной задаче вам следует определить, как долго будет длиться игра, если проигрывающий игрок пытается играть как можно дольше, а выигрывающий игрок пытается закончить как можно раньше, но при этом не делая ходов, которые позволят выиграть его оппоненту. Также вам нужно вывести один из возможных ходов первого игрока, приводящего к описанному результату.

Input

Первая строка содержит одно целое число n ($1 \leq n \leq 10^5$).

Вторая строка содержит n целых чисел a_i ($1 \leq a_i \leq 10^{12}$) — размеры кучек.

Output

В первой строке выведите одно целое число — количество ходов, которые будет длиться игра при оптимальных действиях обоих игроков.

Во второй строке выведите ход первого игрока в формате $i k$, где i — индекс кучки, а k — количество камней, взятых из i -ой кучки. Если существует несколько возможных ходов, выведите любой из них.

Examples

standard input	standard output
2 1 3	3 2 2
2 2 2	4 1 1

Problem E. Guess Table

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 256 mebibytes

Жюри загадало матрицу A размера $n \times m$, состоящую из нулей и единиц. Вам нужно её отгадать. Вы можете сделать не более 200 запросов, после этого вам требуется вывести загаданную матрицу.

Каждый запрос задаётся матрицей B размера $a \times b$ ($1 \leq a \leq n, 1 \leq b \leq m$), состоящей из нулей, единиц и вопросительных знаков. Ответ на запрос — это 1, если в матрице A существует непрерывная подматрица, которая подходит под шаблон B . В противном случае ответ равен 0.

Формально, ответ равен 1, если и только если существуют такие (i, j) ($1 \leq i \leq n - a + 1, 1 \leq j \leq m - b + 1$), что для любых (x, y) ($1 \leq x \leq a, 1 \leq y \leq b$) B_{xy} — это вопросительный знак или $B_{xy} = A_{i+x-1, j+y-1}$.

Также есть еще одно ограничение на матрицу B : в каждой строке и столбце этой матрицы должен быть хотя бы один символ, не являющийся знаком вопроса.

Обратите внимание, что задача интерактивная, поэтому вы должны сбрасывать вывод вашей программы после каждого запроса и после вывода ответа. Для это используйте `fflush(stdout)` в C++, `PrintWriter.flush()` в Java или альтернативу в вашем языке.

Input

В единственной строке заданы два целых числа n и m ($1 \leq n, m \leq 13$) — размер матрицы, которую нужно угадать.

Output

Когда вы готовы вывести ответ, выведите строку `"Answer:"` (без кавычек). После этого, выведите n строк по m символов `"0"` and `"1"` — загаданную матрицу. Не забудьте сбросить вывод вашей программы после этого.

Interaction Protocol

В первой строке каждого запроса выведите строку `"Query:"` (без кавычек) и два целых числа a, b ($1 \leq a \leq n, 1 \leq b \leq m$) — размер подматрицы. После этого выведите a строк по b символов `"0"`, `"1"` или `"?"`. Не забудьте сбросить вывод вашей программы после этого.

После каждого запроса вы получите ответ — одну строку с 1, если загаданная подматрица содержит данную подматрицу и 0 иначе.

Example

standard input	standard output
2 3	Query: 2 1 0 0
1	Query: 1 2 01
1	Query: 2 2 01 0?
1	Query: 2 3 101 ??0
0	Answer: 101 001

Problem H. MIPT Campus

Input file: *standard input*
Output file: *standard output*
Time limit: 3 seconds (5 seconds for Java)
Memory limit: 256 mebibytes

Часть Долгопрудного, относящаяся к физтеху, очень удобна для студентов. Длинная улица Первомайская делит Долгопрудный на две части: учебные здания слева от дороги и общежития справа. Введем ось OX параллельно этой улице. Ширина дороги и расстояния от зданий до дороги незначительны, так что можно считать данную модель одномерной.

На физтехе ровно n студентов, которые посещают занятия. Каждое утро все студенты идут из их общежитий в их учебный корпус, пересекая Первомайскую улицу. i -ый студент живет в общежитии с x -координатой a_i и учится в здании с x -координатой b_i . Студенты могут пересекать улицу только по пешеходным переходам. Уже построено m переходов, перпендикулярных оси OX , расположенных в x -координатах c_1, c_2, \dots, c_m . Студенты всегда выбирают самый короткий путь, чтобы дойти от общежития до учебного корпуса.

Недавно молодежный комитет МФТИ решил, что количество пешеходных переходов недостаточно для студентов. Он пролоббировал инициативу постройки еще одного перехода через любую точку Первомайской. Сейчас его задача — выбрать место постройки так, чтобы минимизировать сумму расстояний, которые проходят студенты от общежитий до учебных зданий. Вам нужно найти эту минимальную сумму расстояний.

Input

Первая строка содержит одно целое число n ($1 \leq n \leq 5 \cdot 10^5$) — количество студентов МФТИ, которые посещают занятия.

Каждая из следующих n строк содержит по два целых числа a_i, b_i ($1 \leq a_i, b_i \leq 10^9$) — координаты общежития и учебного корпуса i -го студента.

Следующая строка содержит одно целое число m ($0 \leq m \leq 5 \cdot 10^5$) — количество переходов.

Последняя строка содержит m различных целых чисел c_1, c_2, \dots, c_m ($1 \leq c_i \leq 10^9$) — координаты переходов.

Output

Выведите одно целое число — минимальную сумму расстояний, которые студенты должны будут проходить после постройки нового перехода. Можно доказать, что это число всегда целое.

Examples

standard input	standard output
2 5 10 20 30 0	35
2 5 10 10 20 1 2	15
2 5 10 20 30 1 11	17

Problem J. Total control

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 256 mebibytes

Город N представляет из себя выпуклый многоугольник на плоскости.

В случае, если на вооружении стражников города N винтовки, стреляющие на расстояние d ($d \geq 0$), князь города считает подконтрольными ему точки внутри города, а также точки на расстоянии не более d от города.

Самолюбие князя будет удовлетворено, если ему будет подконтрольна территория площади как минимум S . Какова минимальная дальнострельность винтовок, которыми нужно экипировать стражников для этого?

Input

В первой строке задано два целых числа n и S ($3 \leq n \leq 5 \cdot 10^4, 1 \leq S \leq 10^{13}$) — количество вершин многоугольника, задающего город и необходимая площадь.

В каждой из следующих n строк задано по два целых числа x и y ($-10^6 \leq x, y \leq 10^6$) — координаты очередной вершины многоугольника.

Гарантируется, что заданные n точек образуют обход выпуклого n -угольника против часовой стрелки.

Output

Выведите единственное вещественное число — минимальную дальнострельность винтовок. Ответ будет считаться правильным, если он имеет относительную или абсолютную погрешность не более 10^{-6} относительно правильного ответа.

Example

standard input	standard output
4 2 0 0 1 0 1 1 0 1	0.21402387849518847

Problem K. Möbius

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 256 mebibytes

Для целого положительного числа n определим функцию Мёбиуса $\mu(n)$. Она принимает значения в $\{-1, 0, 1\}$ и зависит от разложения n на простые числа:

- $\mu(x) = 1$, если x свободно от квадратов и имеет четное число простых делителей.
- $\mu(x) = -1$, если x свободно от квадратов и имеет нечетное число простых делителей.
- $\mu(x) = 0$, если x делится на квадрат какого-то простого.

Например, $\mu(1) = 1, \mu(2) = -1, \mu(6) = 1, \mu(12) = 0$.

Даны два массива a и b , содержащие положительные целые числа.

Обозначим k_y количество пар $(i, j), 1 \leq i \leq n, 1 \leq j \leq m$ таких, что $\mu(a_i \cdot b_j)$ равно y .

Ваша задача состоит в подсчете k_{-1}, k_0 и k_1 .

Input

В первой строке заданы два целых числа n, m ($1 \leq n, m \leq 2 \cdot 10^5$) — размеры массивов a и b .

Вторая строка содержит n целых чисел a_i ($1 \leq a_i \leq 10^6$), разделенных пробелами.

Третья строка содержит m целых чисел b_i ($1 \leq b_i \leq 10^6$), разделенных пробелами.

Output

Выведите три целых числа k_{-1}, k_0, k_1 , разделенных пробелами, в единственной строке, где $k_y = |\{(i, j) : \mu(a_i \cdot b_j) = y\}|$.

Example

standard input	standard output
6 4 1 2 3 4 5 6 2 3 5 7	6 9 9

Problem L. Continuous Fraction

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 512 mebibytes

Вам дана формула

$$a_1 + \frac{b_1}{a_2 + \frac{b_2}{a_3 + \frac{b_3}{\dots + \frac{b_n}{a_{n-1} + \frac{b_n}{a_n}}}}}$$

и значения a_i и b_i . Вычислите результат.

Input

Первая строка входа содержит одно целое число T ($1 \leq T \leq 500$) — количество тестовых примеров.

Первая строка каждого тестового примера содержит одно целое число n ($n \leq 8$). Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10$). Третья строка содержит n целых чисел b_1, b_2, \dots, b_n ($1 \leq b_i \leq 10$).

Output

Для каждого тестового примера в отдельной строке выведите результат вычислений в виде несократимой дроби p/q . В частности, если результат целый, $q = 1$.

Example

standard input	standard output
1	1/2
2	
1 1	
2 3	

Problem M. Morse Alphabet

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 256 mebibytes

В этой задаче требуется написать программу, которая бы декодировала сообщения, заданные азбукой Морзе. Таблица кодирования латинских букв в азбуке Морзе приведена ниже.

A	.-	B	-...	C	-.-. .	D	-..	E	.	F	...-
G	--.	H	I	..	J	.----	K	--.	L	...-
M	--	N	-. .	O	---	P	...-	Q	---.	R	..-
S	...	T	-	U	...-	V-	W	...-	X	...-
Y	..--	Z	--..								

Если передаются несколько букв, они разделяются символом ('/'). Пауза между словами обозначается двумя символами ('/').

Input

Входной файл состоит из не более чем 1234 тестовых примеров. Каждый тестовый пример задан в отдельной строке и имеет длину не более 1000 символов. Кроме того, он удовлетворяет следующим ограничениям:

- Строка содержит только символы ('-'), ('.'), и ('/').
- Строка пуста.
- Первый и последний символы отличны от '/'.
• В строке нет более, чем двух подряд идущих '/'.
• Каждая непустая подстрока между двумя '/' представляет собой корректный код Морзе.

Output

Для каждого тестового примера в отдельной строке заглавными латинскими буквами выведите декодированное сообщение.

Example

standard input	standard output
../-.../.../...-	ICPC
---/---/---/...//.../...-/.../---	GOOD LUCK

Problem N. Two Sequences

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 512 mebibytes

Вам даны две последовательности a_1, a_2, \dots, a_n и b_1, b_2, \dots, b_m , а также целое число p . Вычислите количество таких чисел q , что последовательность b_1, b_2, \dots, b_m совпадает с последовательностью $a_q, a_{q+p}, a_{q+2p}, \dots, a_{q+(m-1)p}$, где $q + (m-1)p \leq n$ и $q \geq 1$.

Input

Первая строка входа содержит одно целое число $T \leq 100$ — количество тестовых примеров.

Каждый тестовый пример состоит из трёх строк. Первая строка содержит три целых числа $1 \leq n \leq 4 \cdot 10^4$, $1 \leq m \leq 4 \cdot 10^4$ и $1 \leq p \leq 4 \cdot 10^4$. Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$). Третья строка содержит m целых чисел b_1, b_2, \dots, b_m ($1 \leq b_i \leq 10^9$).

Гарантируется, что сумма всех n во входе не превосходит $2 \cdot 10^6$.

Output

Для каждого тестового примера в отдельной строке выведите одно целое число y — ответ к задаче.

Example

standard input	standard output
2	2
6 3 1	1
1 2 3 1 2 3	
1 2 3	
6 3 2	
1 3 2 2 3 1	
1 2 3	

Problem O. Build The Square

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 256 mebibytes

Вам заданы три прямоугольника. Определите, можно ли из них сложить квадрат. При этом поворачивать прямоугольники можно; перекрытия и «отверстия» внутри квадрата запрещены.

Input

Входной файл состоит из трёх строк. В первой строке заданы два целых числа w_1 и h_1 ($1 \leq w_1, h_1 \leq 100$) — длины сторон первого прямоугольника. Во второй и третьей строках в аналогичном формате с аналогичными ограничениями заданы длины сторон второго и третьего прямоугольников.

Output

Если из данных прямоугольников можно без наложений сложить квадрат, выведите “Yes”. Иначе выведите “No”.

Example

standard input	standard output
8 2 1 6 7 6	1

Problem P. No Triangles

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 512 mebibytes

Даны n палочек с длинами $1, 2, 3, \dots, n$. Какое наименьшее количество палочек нужно убрать, чтобы ни из каких трёх оставшихся палочек нельзя было сложить треугольник?

Input

Первая строка содержит одно целое число T — количество тестовых примеров.

Каждый тестовый пример состоит из одной строки, содержащей целое число n ($1 \leq n \leq 10^9$) — количество палочек.

Output

Для каждого тестового примера выведите минимальное количество убираемых палочек.

Example

standard input	standard output
3	0
2	1
4	2
6	