

Problem A. НБП

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

Дана последовательность, требуется найти длину её наибольшей возрастающей подпоследовательности. Подпоследовательностью последовательности называется некоторый набор её элементов, не обязательно стоящих подряд.

Input

В первой строке входных данных задано число N — длина последовательности ($1 \leq N \leq 1000$). Во второй строке задается сама последовательность (разделитель — пробел). Элементы последовательности — целые числа, не превосходящие 10000 по абсолютной величине.

Output

Требуется вывести длину наибольшей строго возрастающей подпоследовательности.

Examples

standard input	standard output
6 3 29 5 5 28 6	3
10 4 8 2 6 2 10 6 29 58 9	5

Problem B. Наибольшая общая подпоследовательность

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second(s)
Memory limit: 512 MiB

Общей подпоследовательностью двух строк s_1 и s_2 называется пара последовательностей индексов $(\{a_i\}, \{b_i\})$ такая, что $a_1 < a_2 < \dots < a_k$, $b_1 < b_2 < \dots < b_k$, and $s_1[a_i] = s_2[b_i]$ for all $1 \leq i \leq k$.

Найдите наибольшую общую подпоследовательность двух строк.

Input

Первая и вторая строки входа содержат две непустые строки, каждая из которых состоит из строчных латинских букв. Длина каждой строки не превосходит 100.

Output

В первой строке выведите целое число k — длину наибольшей общей подпоследовательности. Во второй выведите k целых чисел — индексы символов наибольшей общей подпоследовательности в первой строке, отсортированные по возрастанию. В третьей, аналогично — отсортированные по возрастанию индексы символов наибольшей общей подпоследовательности во второй строке. Символы в строках занумерованы с 1.

Если способов выбрать наибольшую общую подпоследовательность несколько, выведите любой из них.

Example

standard input	standard output
abcd	2
cxbydz	3 4
	1 5

Problem C. Шаблоны

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

Многие операционные системы используют шаблоны для ссылки на группы объектов: файлов, пользователей, и т. д. Ваша задача — реализовать простейший алгоритм проверки шаблонов для имен файлов.

В этой задаче алфавит состоит из маленьких букв английского алфавита и точки ('.'). Шаблоны могут содержать произвольные символы алфавита, а также два специальных символа: '?' и '*'. Знак вопроса '?' соответствует ровно одному произвольному символу. Звездочка '*' соответствует подстроке произвольной длины (возможно, нулевой). Символы алфавита, встречающиеся в шаблоне, отображаются на ровно один такой же символ в проверяемой строке. Строка считается подходящей под шаблон, если символы шаблона можно последовательно отобразить на символы строки таким образом, как описано выше. Например, строки "ab", "aab" и "beda." подходят под шаблон "*a?", а строки "bebe", "a" и "ba" — нет.

Input

Первая строка входного файла определяет шаблон P . Вторая строка S состоит только из символов алфавита. Ее необходимо проверить на соответствие шаблону. Длины обеих строк не превосходят 10 000. Строки могут быть пустыми — будьте внимательны!

Output

Если данная строка подходит под шаблон, выведите YES. Иначе выведите NO.

Examples

standard input	standard output
k?t*n kitten	YES
k?t?n kitten	NO

Problem D. Редакционное расстояние

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

В информатике *редакционным расстоянием* между двумя строками называется минимальное количество добавлений, удалений и замен символов, при помощи которых можно из одной строки получить другую. К примеру, редакционное расстояние между строками “ab” и “ab” равно нулю, так как строки равны между собой безо всяких изменений; расстояние между строками “short” и “ports” равно трём: в слове “short” нужно удалить из начала букву ‘s’, заменить ‘h’ на ‘p’ и добавить в конец букву ‘s’. Редакционное расстояние также называют *расстоянием Левенштейна*.

Найдите редакционное расстояние между двумя заданными строками.

Input

В первой строчке входного файла задана одна строка, во второй — другая.

Длины строк от 1 до 100.

Output

В выходной файл выведите единственное число — редакционное расстояние между двумя заданными строками.

Examples

standard input	standard output
ab ab	0
short ports	3

Problem E. Наибольшая пилообразная подпоследовательность

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

Числовая последовательность называется пилообразной если каждый ее член (кроме первого и последнего) либо больше обоих своих соседей, либо меньше обоих соседей. Например, последовательность 1, 2, 1, 3, 2 является пилообразной, а 1, 2, 3, 1, 2 — нет, поскольку $1 < 2 < 3$. Любая последовательность из одного элемента является пилообразной. Последовательность из двух элементов является пилообразной, если ее элементы не равны.

Дана последовательность. Требуется определить, какое наименьшее количество ее членов нужно вычеркнуть, чтобы оставшаяся последовательность оказалась пилообразной.

Input

В первой строке входного файла записано одно число N ($1 \leq N \leq 10^5$) — количество членов последовательности. Во второй строке записано N натуральных чисел, не превосходящих 10^4 — члены последовательности.

Output

В выходной файл выведите одно число — минимальное количество членов, которые необходимо вычеркнуть.

Examples

standard input	standard output
5 1 2 3 1 2	1
5 1 2 1 3 2	0

Problem F. Гангстеры

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

N гангстеров собираются в ресторан. i -й гангстер приходит в момент времени T_i и имеет богатство P_i .

Дверь ресторана имеет $K + 1$ степень открытости, они обозначаются целыми числами из интервала $[0, K]$. Степень открытости двери может изменяться на единицу в единицу времени, то есть дверь может открыться на единицу, закрыться на единицу или остаться в том же состоянии. В начальный момент времени дверь закрыта (степень открытости 0).

i -й гангстер заходит в ресторан, только если дверь открыта специально для него, то есть когда степень открытости двери соответствует его полноте S_i . Если в момент, когда гангстер подходит к ресторану, степень открытости двери не соответствует его полноте, он уходит и больше не возвращается. Ресторан работает в интервале времени $[0, T]$.

Требуется собрать гангстеров с максимальным суммарным богатством в ресторане, открывая и закрывая дверь соответствующим образом.

Input

В первой строке находятся числа N, K, T , во второй - T_1, T_2, \dots, T_N , в третьей - P_1, P_2, \dots, P_N . в четвёртой - S_1, S_2, \dots, S_N . Числа в строках разделены пробелами.

$1 \leq N \leq 100, 1 \leq K \leq 100, 1 \leq T \leq 3 \cdot 10^4, 0 \leq T_i \leq T, 1 \leq P_i \leq 300, 1 \leq S_i \leq K$, все числа целые.

Output

Вывести одно число - максимальное суммарное богатство гангстеров, попавших в ресторан. Если зайти не удалось никому, вывести 0.

Examples

standard input	standard output
2 10 20 10 16 10 11 10 7	21
1 10 20 15 10 10	10

Problem G. Упорядочьте шарики

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second(s)
Memory limit: 512 MiB

При игре в новую игру (некоторый гибрид боулинга и бильярда) используется N шариков, пронумерованных числами от 1 до N . В начале игры эти шарики должны быть выложены в линию в порядке своих номеров. В процессе игры их порядок может меняться.

Для того, чтобы упорядочить шарики перед началом следующей партии, используется следующее устройство. Это устройство состоит из головки, которая, перемещаясь над шариками, может «засасывать» и «выплевывать» шарики. Чтобы получить большее представление об этом устройстве, представьте себе пылесос, который может засасывать шарики, перемещаться в нужное место, и там, включаясь на продув в обратном направлении, шарики «выплевывать».

При засасывании шарика все шарики, которые находились правее засасываемого, сдвигаются влево. «Выплюнуть» шарик можно между любыми двумя шариками (а также перед первым шариком или после последнего), тогда выплевываемый шарик вставляется между этими шариками, и все шарики, которые находятся правее вставляемого, сдвигаются вправо.

В устройство может быть одновременно засосано больше одного шарика, при этом при выплевывании шарика первым выплевывается последний засосанный шарик, затем - предпоследний и т.д. (т.е. устройство работает по принципу стека). Шарики выплевываются по одному, т.е. можно выплюнуть только один шарик, остальные оставив внутри устройства (при этом дальше можно как продолжать «выплевывать» шарики в том же или в другом месте, так и засасывать новые шарики).

Наиболее энергоемкой из описанных операций является операция засасывания шарика, поэтому хочется минимизировать количество именно таких операций.

Напишите программу, которая по данному начальному расположению шариков определит минимальное количество операций засасывания, которое нужно, чтобы расположить шарики в порядке их номеров.

Input

Во входном файле задано сначала число N — количество шариков ($1 \leq N \leq 1000$). Далее идет N чисел, задающих номера шариков в порядке слева направо в их текущем расположении (каждое число — от 1 до N , и каждое из чисел встречается в последовательности ровно один раз).

Output

В выходной файл выведите одно число — минимальное количество операций засасывания шарика, которое потребуется, чтобы расположить шарики в порядке их номеров.

Examples

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

Problem A. Префикс-функция

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second(s)
Memory limit: 512 MiB

Дана непустая строка S , длина которой N не превышает 10^6 . Будем считать, что элементы строки нумеруются от 1 до N .

Требуется для всех i от 1 до N вычислить её префикс-функцию $\pi[i]$.

Input

Одна строка длины N , $0 < N \leq 10^6$, состоящая из маленьких латинских букв.

Output

Выведите N чисел — значения префикс-функции для каждой позиции, разделённые пробелом.

Example

standard input	standard output
abracadabra	0 0 0 1 0 1 0 1 2 3 4

Problem B. Z-функция

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second(s)
Memory limit: 512 MiB

Дана непустая строка S , длина которой N не превышает 10^6 . Будем считать, что элементы строки нумеруются от 1 до N .

Требуется для всех i от 1 до N вычислить её z-функцию $z[i]$.

Input

Одна строка длины N , $0 < N \leq 10^6$, состоящая из маленьких латинских букв.

Output

Выведите N чисел — значения z-функции для каждой позиции, разделённые пробелом.

Examples

standard input	standard output
abracadabra	11 0 0 1 0 1 0 4 0 0 1

Problem C. Сравнения подстрок

Input file: *standard input*
Output file: *standard output*
Time limit: 2 second(s)
Memory limit: 512 mebibytes

Дана строка. Нужно уметь отвечать на запросы вида: равны ли подстроки $[a..b]$ и $[c..d]$.

Input

Сперва строка S (не более 10^5 строчных латинских букв). Далее число M — количество запросов.

В следующих M строках запросы a, b, c, d . $0 \leq M \leq 10^5, 1 \leq a \leq b \leq |S|, 1 \leq c \leq d \leq |S|$

Output

M строк. Выведите Yes, если подстроки совпадают, и No иначе.

Examples

standard input	standard output
trololo	Yes
3	Yes
1 7 1 7	No
3 5 5 7	
1 1 1 5	

Problem D. Дана строка

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

Даже больше — дано две строки, α и β . Вам требуется узнать, где в строке α можно найти строку β как подстроку и выписать все такие позиции.

Input

В первой строке входного файла содержится строка α , во второй — строка β . Строки состоят только из строчных латинских букв (a–z), их длины не превосходят 100 000.

Output

В первой строке выходного файла выведите одно число — количество вхождений строки β в строку α . Во второй строке для каждого вхождения выведите номер символа в строке α , где начинается очередная строка β . Вхождения нужно выводить в возрастающем порядке.

Examples

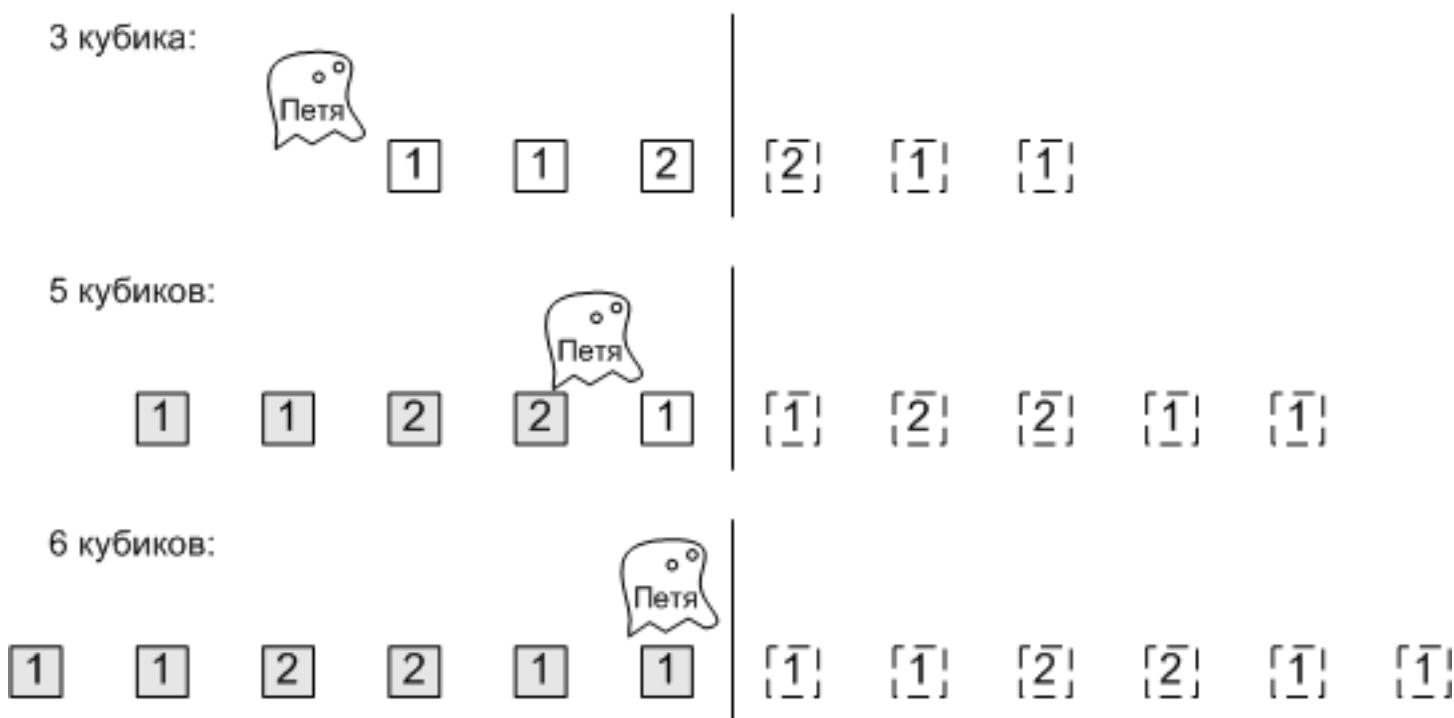
standard input	standard output
abacaba	2
aba	1 5

Problem E. Кубики

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

Привидение Петя любит играть со своими кубиками. Он любит выкладывать их в ряд и разглядывать своё творение. Однако недавно друзья решили подшутить над Петей и поставили в его игровой комнате зеркало. Ведь всем известно, что привидения не отражаются в зеркале! А кубики отражаются.

Теперь Петя видит перед собой N цветных кубиков, но не знает, какие из этих кубиков настоящие, а какие — всего лишь отражение в зеркале.



Помогите Пете! Выясните, сколько у него может быть кубиков. Петя видит отражение всех кубиков в зеркале и часть кубиков, которая находится перед ним. Часть кубиков может быть позади Пети, их он не видит.

Input

Первая строка входного файла содержит два целых числа: N ($1 \leq N \leq 100\,000$) и количество различных цветов, в которые могут быть раскрашены кубики, — M ($1 \leq M \leq 100\,000$). Следующая строка содержит N целых чисел от 1 до M — цвета кубиков.

Output

В выходной файл выведите в порядке возрастания все такие K , что у Пети может быть K кубиков.

Example

standard input	standard output
6 2 1 1 2 2 1 1	3 5 6

Problem F. Основание строки

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

Строка S была записана много раз подряд, после чего из получившейся строки взяли подстроку и дали вам. Ваша задача определить минимально возможную длину исходной строки S .

Input

В первой и единственной строке входного файла записана строка, которая содержит только латинские буквы, длина строки не превышает 50 000 символов.

Output

В выходной файл выведите ответ на задачу.

Examples

standard input	standard output
zzz	1
bcabcab	3

Problem G. Retrostring

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

S называется последовательность символов S_1, \dots, S_n , где $|S| = n$ — это *длина* строки S .

Для любого k ($1 \leq k \leq |S|$) k -м *префиксом* строки S называется строка S_1, \dots, S_k длины k . Если $k < |S|$, то префикс называется *собственным*.

Аналогично для любого k ($1 \leq k \leq |S|$) k -м *суффиксом* строки S называется строка $S_{|S|-k+1}, \dots, S_{|S|}$ длины k . Если $k < |S|$, то суффикс также называется *собственным*.

Назовём *числом повторяемости* строки S количество её различных собственных суффиксов, каждый из которых совпадает с префиксом той же длины, что и этот суффикс.

Назовём строку *ретрострокой*, если её число повторяемости строго больше чисел повторяемости всех её собственных префиксов.

Дана строка S . Нужно найти её префикс максимальной длины (не обязательно собственный), являющийся ретрострокой.

Input

В первой строке входного файла записана строка S , $1 \leq |S| \leq 1000000$. Строка содержит лишь символы с ASCII-кодом от 33 до 126.

Output

В первой строке выходного файла должен быть выведен префикс S максимальной длины, являющийся ретрострокой.

Examples

standard input	standard output
z	z
aabaabaaabaabaabaaba	aabaabaaabaabaa

Problem H. Шифр Бэкона

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second(s)
Memory limit: 512 MiB

Программисту Васе не повезло — вместо отпуска его послали в командировку на научную конференцию. «Надо повышать уровень знаний», — сказал начальник, «Важная конференция по криптографии, проводится во Франции — а там шифровали еще во времена Ришелье и взламывали чужие шифры еще во времена Виета.»

Вася быстро выяснил, что все луврские картины он уже где-то видел, вид Эйфелевой башни приелся ему еще раньше, чем мышка стерла его с коврика, а такие стеклянные пирамиды у нас делают надо всякими киосками и сомнительными забегаловками. Одним словом, смотреть в Париже оказалось просто не на что, рыбу половить негде, поэтому Васе пришлось посещать доклады на конференции.

Один из докладчиков, в очередной раз пытаясь разгадать шифры Бэкона, выдвинул гипотезу, что ключ к тайнам Бэкона можно подобрать, проанализировав все возможные подстроки произведений Бэкона. «Но их же слишком много!» — вслух удивился Вася. «Нет, не так уж и много!» — закричал докладчик, — «Подсчитайте, и вы сами убедитесь!»

Тем же вечером Вася нашел в интернете полное собрание сочинений Бэкона. Он написал программу, которая переработала тексты в одну длинную строку, выкинув из текстов все пробелы и знаки препинания. И вот теперь Вася весьма озадачен — а как же подсчитать количество различных подстрок этой строки?

Input

На входе дана непустая строка, полученная Васей. Строка состоит только из строчных латинских символов. Ее длина не превосходит 2 000 символов.

Output

Выведите количество различных подстрок этой строки.

Examples

standard input	standard output
aaba	8

Problem I. Палиндромы

Input file: *standard input*
Output file: *standard output*
Time limit: 2 second
Memory limit: 512 MiB

Строка называется палиндромом, если она одинаково читается как слева направо, так и справа налево. Например, *abba* — палиндром, а *otax* — нет. Для строки α будем обозначать $\alpha_{i,j}$ ее подстроку длины $j - i + 1$ с i -ой позиции по j -ую включительно. Для заданной строки длины N ($1 \leq N \leq 100000$) требуется подсчитать число пар $(i; j)$, $1 \leq i < j \leq N$ таких, что $\alpha_{i,j}$ является палиндромом.

Input

Входной файл содержит одну строку α длины N , состоящую из маленьких латинских букв.

Output

В выходной файл выведите искомое число — количество палиндромов.

Examples

standard input	standard output
aaa	3
abba	2

Problem J. Преобразование строковых функций

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

Для строки S определим Z -функцию следующим образом: $Z[i] = \text{lcp}(S, S[i..|S|])$, где $\text{lcp}(S_1, S_2)$ равно длине наибольшего общего префикса строк S_1 и S_2 . Например, для $S = abacabaa$ Z -функция равна $[8, 0, 1, 0, 3, 0, 1, 1]$.

Для строки S определим ее префикс-функцию: $\pi[i] = \max\{k | 0 \leq k < i, S[1..k] = S[i - k + 1..i]\}$. Например, для $S = abacabaa$ ее префикс-функция имеет вид: $[0, 0, 1, 0, 1, 2, 3, 1]$.

Для некоторой строки S была посчитана ее Z -функция, а строка S была утеряна. Ваша задача получить ее префикс-функцию по заданной Z -функции.

Input

В первой строке входного файла содержится натуральное число N ($1 \leq N \leq 200\,000$), где N — длина S . Во второй строке записана Z -функция строки S .

Output

Выведите N чисел — искомую префикс-функцию.

Examples

standard input	standard output
8	0 0 1 0 1 2 3 1
8 0 1 0 3 0 1 1	

Problem К. Обратное преобразование строковых функций

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

Для строки S определим Z -функцию следующим образом: $Z[i] = lcp(S, S[i..|S|])$, где $lcp(S1, S2)$ равно длине наибольшего общего префикса строк $S1$ и $S2$. Например, для $S = abacabaa$ Z -функция равна $[8, 0, 1, 0, 3, 0, 1, 1]$.

Для строки S определим ее префикс-функцию: $\pi[i] = \max\{k | 0 \leq k < i, S[1..k] = S[i - k + 1..i]\}$. Например, для $S = abacabaa$ ее префикс-функция имеет вид: $[0, 0, 1, 0, 1, 2, 3, 1]$. Для некоторой строки была посчитана ее префикс-функция, а строка была утеряна. Ваша задача получить ее Z -функцию по заданной префикс-функции.

Input

В первой строке входного файла содержится натуральное число N ($1 \leq N \leq 200000$), где N — длина S . Во второй строке записана префикс-функция строки S .

Output

Выведите N чисел — искомую Z -функцию.

Example

standard input	standard output
8 0 0 1 0 1 2 3 1	8 0 1 0 3 0 1 1

Problem A. Длина вектора

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

Дан вектор, найдите его длину.

Input

Входные данные содержат четыре целых числа x_0 , y_0 , x_1 , y_1 , по абсолютной величине не превосходящие 10^4 — координаты начала и конца вектора.

Output

Выведите длину вектора с абсолютной погрешностью не хуже 10^{-4} .

Example

standard input	standard output
1 1 2 2	1.41421356237309500

Problem B. Угол между векторами

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second(s)
Memory limit: 512 MiB

Даны два вектора. Найдите полярный угол между ними.

Input

Четыре числа — координаты двух ненулевых векторов.

Output

Одно число — величина неориентированного угла между ними с точностью до седьмого знака после запятой из интервала $[0, \pi]$.

Examples

standard input	standard output
2 1 3 5	0.56672921752350635000

Problem C. Расстояние от точки до отрезка

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second(s)
Memory limit: 512 MiB

Найдите расстояние от заданной точки до заданного отрезка.

Input

Шесть целых чисел — координаты точки и координаты концов отрезка.

Output

Одно число — расстояние от точки до отрезка с точностью не менее 10^{-6} .

Example

standard input	standard output
0 4 2 3 2 5	2.000000000000000000000000

Problem D. Проверка пересечения отрезков

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second(s)
Memory limit: 512 MiB

Даны два отрезка. Требуется выяснить, пересекаются ли они.

Input

Восемь чисел – координаты концов двух отрезков, по модулю не превосходящие 10^4 .

Output

Одна строка “YES”, если отрезки имеют общие точки, и “NO” в противном случае.

Example

standard input	standard output
1 2 1 2 1 2 1 2	YES
3 3 5 6 5 6 3 3	YES

Problem E. Точка пересечения медиан

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second(s)
Memory limit: 512 MiB

Дан невырожденный треугольник. Найти координаты точки пересечения медиан.

Input

Вход содержит координаты трех точек, не лежащих на одной прямой: $X_1, Y_1, X_2, Y_2, X_3, Y_3$. Все числа целые, по модулю не превосходят 10^6 .

Output

Выведите два числа X и Y , задающие координаты точки пересечения медиан в треугольнике, образованном исходными точками, с точностью не менее 5 знаков после десятичной точки.

Example

standard input	standard output
1 1 2 4 3 2	2.0 2.33333

Problem F. Точка пересечения биссектрис

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second(s)
Memory limit: 512 MiB

Дан невырожденный треугольник. Найти координаты точки пересечения биссектрис.

Input

Вход содержит координаты трех точек, не лежащих на одной прямой: $X_1, Y_1, X_2, Y_2, X_3, Y_3$. Все числа целые, по модулю не превосходят 1000.

Output

Выведите два числа X и Y , задающие координаты точки пересечения биссектрис в треугольнике, образованном исходными точками, с точностью не менее 5 знаков после десятичной точки.

Example

standard input	standard output
1 1 2 4 3 2	2.12132034355964 2.29289321881345

Problem G. Площадь многоугольника

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second(s)
Memory limit: 512 MiB

Задан простой многоугольник (не обязательно выпуклый!). Найдите его площадь.

Input

В первой строке одно число N ($3 \leq N \leq 100\,000$). Далее в N строках по паре чисел — координаты очередной вершины простого многоугольника в порядке обхода по или против часовой стрелки.

Координаты целые по модулю не превосходят 10 000.

Output

Одно число — величина площади приведённого многоугольника с абсолютной точностью.

Examples

standard input	standard output
3 1 0 0 1 1 1	0.5

Problem H. Треугольник и точка

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second(s)
Memory limit: 512 MiB

В декартовой системе координат на плоскости заданы координаты вершин треугольника и ещё одной точки. Определить, принадлежит ли эта точка треугольнику.

Input

В четырёх строках находятся пары чисел — координаты точек. Числа в первых трёх строках — это координаты вершин треугольника, в четвёртой строке — координаты тестируемой точки. Координаты вершин — целые числа, для любой точки выполняются следующие условия: $-10^4 \leq x, y \leq 10^4$.

Output

Вывести слово “In”, если точка находится внутри или на границе треугольника, или “Out” - если снаружи.

Examples

standard input	standard output
-2 -2 3 1 0 1 0 0	In
-2 -2 3 1 0 1 2 -2	Out

Problem I. Проверка на выпуклость

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second(s)
Memory limit: 512 MiB

Дан многоугольник. Проверить, является ли он строго выпуклым (в строго выпуклом многоугольнике, в частности, никакие три вершины не коллинеарны).

Input

Первая строка входных данных содержит одно целое число n ($3 \leq n \leq 10^5$) — количество вершин многоугольника. Каждая из последующих n строк содержит два целых числа x_i и y_i ($-10^9 \leq x_i, y_i \leq 10^9$), задающих координаты i -й вершины в порядке обхода.

Output

Выведите **Yes**, если многоугольник выпуклый, и **No** в противном случае.

Examples

standard input	standard output
4 1 1 1 3 3 3 3 1	Yes
3 1 0 1 1 0 2	Yes

Problem J. Точка пересечения прямых

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second(s)
Memory limit: 512 MiB

На плоскости даны две прямые. Каждая прямая задается парой точек, через которые она проходит. Требуется установить, пересекаются ли эти прямые, и найти координаты точки пересечения.

Input

Вводятся сначала координаты двух различных точек, через которые проходит первая прямая, а затем — координаты еще двух различных (но, быть может, совпадающих с первыми двумя) точек, через которые проходит вторая прямая. Координаты каждой точки - целые числа, по модулю не превышающие 1000.

Output

Если прямые не пересекаются, выведите одно число 0. Если прямые совпадают, выведите 2. Если прямые пересекаются ровно в одной точке, то выведите сначала число 1, а затем два вещественных числа — координаты точки пересечения. Допустимая абсолютная погрешность — 10^{-2} .

Example

standard input	standard output
0 0 1 1 1 0 -1 2	1 0.5 0.5

Problem K. Окружность и прямая

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second(s)
Memory limit: 512 MiB

Даны окружность и прямая. Найти все точки их пересечения.

Input

Шесть целых чисел — координаты центра X_i и Y_i , радиус окружности R_i и коэффициенты A , B и C уравнения прямой ($0 \leq R_i \leq 10^4$, все координаты не превосходят 10^4 по абсолютной величине).

Output

В первой строке одно число K , равное количеству точек пересечения прямой с окружностью. Далее в K строках координаты самих точек с точностью не хуже 10^{-3} . Точки отсортировать по возрастанию x , а затем по возрастанию y .

Examples

standard input	standard output
2 4 1 1 -1 0	0
2 3 1 1 -1 0	2 2.00000 2.00000 3.00000 3.00000

Problem L. Две окружности

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second(s)
Memory limit: 512 MiB

Вам даны две окружности в плоскости. Найдите все их различные точки пересечения.

Input

В первой строке следует целое число M ($1 \leq M \leq 10\,000$) — количество тестов.

Следующие $2 \cdot M$ строк описывают сами окружности. Каждая окружность описывается тремя целыми числами: координатами центра и радиусом.

Гарантируется, что все координаты от $-1\,000$ до $1\,000$, а все радиусы окружностей положительны и не превосходят $1\,000$.

Output

Для каждого теста выведите следующее.

Если у окружностей нет точек пересечения, выведите 0.

Если у окружностей бесконечное количество точек пересечения, выведите -1.

Если у окружностей есть точки пересечения и их количество конечно, выведите целое число k — количество точек пересечения, а в следующие k строк выведите сами точки пересечения, сортируя их по x в первую очередь и по y во вторую. Абсолютная или относительная погрешность выведенных точек пересечения не должна превышать 10^{-8} .

Разделяйте выводы для разных тестов пустой строкой.

Examples

standard input	standard output
1 0 0 2 4 0 2	1 2.00000000000000000000 0.00000000000000000000
2 0 0 2 4 0 2 0 0 1 1000 1000 1	1 2.00000000000000000000 0.00000000000000000000 0

Problem A. Выход из лабиринта

Input file: *standard input*
Output file: *standard output*
Time limit: 2 second(s)
Memory limit: 512 MiB

Во время торнадо в Костромской области было повалено большое количество деревьев. Однажды утром, спустя пару дней после торнадо, мальчик Илья вышел прогуляться и увидел, что поваленные деревья образовали настоящий лабиринт. Вася очень обрадовался, стал гулять и играть в лабиринте. Внезапно ему позвонил преподаватель и поинтересовался, почему он давно не видел Илью. Чтобы не получить дырку в бэйджик за самовольный выход за территорию лагеря, Илья, конечно же, хотел попасть в домик как можно скорее, но и побродить по лабиринту ему тоже очень хотелось. Помогите ему узнать, как быстро он сможет добраться до домика.

Input

В первой строке входного файла заданы целые положительные числа N и M ($1 \leq N, M \leq 1000$). В следующих N строках заданы по M символов, описывающих лабиринт. На позициях, по которым Илья может перемещаться, записан символ «0», а на позициях, которые перегорожены деревьями, записан символ «1». После описания лабиринта следуют целые числа x_1, y_1, x_2, y_2 — координаты (то есть соответствующие номера столбцов и строк) Ильи и его домика соответственно ($1 \leq x_1, x_2 \leq M$, $1 \leq y_1, y_2 \leq N$).

Output

В выходной файл выведите единственное число — длину кратчайшего пути от точки (x_1, y_1) до точки (x_2, y_2) , если Илья может добраться до дому по лабиринту, и «-1» в противном случае.

Examples

standard input	standard output
4 6 1 0 1 1 1 0 0 0 1 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 2 1 5 3	7
4 6 1 0 1 1 1 0 0 0 1 0 0 0 1 0 1 1 0 0 0 0 0 1 0 0 2 1 5 3	-1

Problem B. Island. Островные государства

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

Суровые феодальные времена переживала некогда великая островная страна Байтландия. За главенство над всем островом борются два самых сильных барона. Таким образом, каждый город страны контролируется одним из правителей. Как водится издревле, некоторые из городов соединены двусторонними дорогами. Бароны очень не любят друг друга и стараются делать как можно больше пакостей. В частности, теперь для того чтобы пройти по дороге, соединяющей города различных правителей, надо заплатить пошлину — один байтландский рубль.

Программист Вася живет в городе номер 1. С наступлением лета он собирается съездить в город N на Всебайтландское собрание программистов. Разумеется, он хочет затратить при этом как можно меньше денег и помочь ему здесь, как обычно, предлагается Вам.

Input

В первой строке входного файла записано два числа N и M ($1 \leq N, M \leq 100\,000$) — количество городов и количество дорог соответственно.

В следующей строке содержится информация о городах — N чисел 1 или 2 — какому из баронов принадлежит соответствующий город.

В последних M строках записаны пары $1 \leq a, b \leq N, a \neq b$. Каждая пара означает наличие дороги из города a в город b . По дорогам Байтландии можно двигаться в любом направлении.

Output

Если искомого пути не существует, выведите единственное слово **impossible**. В противном случае в первой строке напишите минимальную стоимость и количество посещенных городов, а во вторую выведите эти города в порядке посещения. Если минимальных путей несколько, выведите любой.

Examples

standard input	standard output
7 8 1 1 1 1 2 2 1 1 2 2 5 2 3 5 4 4 3 4 7 1 6 6 7	0 5 1 2 3 4 7
5 5 1 2 1 1 2 1 2 2 3 3 5 1 4 4 5	1 3 1 4 5

Problem C. Свинки-копилки

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second(s)
Memory limit: 512 MiB

У Васи есть N свинок-копилков, свинки занумерованы числами от 1 до N . Каждая копилка может быть открыта единственным соответствующим ей ключом или разбита.

Вася положил ключи в некоторые из копилков (он помнит, какой ключ лежит в какой из копилков). Теперь Вася собрался купить машину, а для этого ему нужно достать деньги из всех копилков. При этом он хочет разбить как можно меньшее количество копилков (ведь ему еще нужно копить деньги на квартиру, дачу, вертолет. . .). Помогите Васе определить, какое минимальное количество копилков нужно разбить.

Input

В первой строке содержится число N — количество свинок-копилков ($1 \leq N \leq 100$). Далее идет N строк с описанием того, где лежит ключ от какой копилки: в i -ой из этих строк записан номер копилки, в которой находится ключ от i -ой копилки.

Output

Выведите единственное число: минимальное количество копилков, которые необходимо разбить.

Examples

standard input	standard output
4 2 1 2 4	2

Problem D. Экскурсия

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

Школьники собираются на экскурсию в Кострому. Кострома — красивый старый город, в котором площади соединяются друг с другом короткими улицами, каждую из которых можно пройти не более чем за десять минут. Школьники хотят составить интересный маршрут экскурсии. Так как они поедут на автобусах, то маршрут экскурсии должен начинаться и заканчиваться на одной и той же площади. К сожалению, у школьников будет очень мало времени. Поэтому они решили выбрать наиболее короткий кольцевой маршрут, не проходящий ни по какой улице дважды.

Помогите школьникам найти такой маршрут.

Input

Первая строка входного файла содержит два натуральных числа n и m — количество площадей и улиц в Костроме ($1 \leq n \leq 1\,000$, $1 \leq m \leq 10\,000$). Площади занумерованы от 1 до n .

Последующие m строк содержат описания улиц. Каждая улица описывается тремя целыми числами — номерами площадей, которые она соединяет, и количеством минут, которые требуются школьникам на то, чтобы пройти по ней (от одной до десяти минут). Между двумя площадями может быть более одной улицы. Улица соединяет две различные площади.

Гарантируется, что в Костроме существует как минимум один кольцевой маршрут.

Output

Первая строка выходного файла должна содержать единственное число — продолжительность минимального маршрута в минутах.

Example

standard input	standard output
5 6 1 2 1 2 3 10 1 3 1 2 4 1 3 4 1 1 5 1	4

Problem E. Кратчайший путь

Input file: *standard input*
Output file: *standard output*
Time limit: 1 секунда
Memory limit: 512 мегабайт

Надеюсь, все вы умеете искать в ориентированном графе кратчайший путь. В этой задаче вам предлагается свое умение продемонстрировать.

Вам дан ориентированный взвешенный граф. Веса ребер — целые числа от 1000 до 2000. Нужно несколько раз (не более 1000) ответить на следующий запрос: длина кратчайшего пути из некоторой вершины s в некоторую вершину t .

Input

На первой строке числа N и M ($1 \leq N \leq 25\,000$, $0 \leq M \leq 50\,000$) — количество вершин и ребер нашего графа, соответственно. Вершины нумеруются целыми числами от 1 до N . Далее M строк содержат информацию о ребрах графа. Каждое ребро задается тремя числами — номер начала, номер конца и вес. Все веса — целые числа от 1000 до 2000. В графе могут быть и петли, и кратные ребра. Следующая строка содержит число K ($1 \leq K \leq 1000$) — количество запросов. В следующих K строках задаются запросы. Каждый запрос описывается двумя числами — из какой вершины, и в какую должен вести путь.

Output

Для каждого запроса выведите на отдельной строке целое число — длину кратчайшего пути. Если кратчайшего пути не существует следует вывести -1 .

Example

standard input	standard output
5 5	-1
1 2 2000	3000
1 3 1000	0
1 4 1200	2000
2 3 1500	
3 4 1500	
4	
1 5	
2 4	
3 3	
1 2	

Note

Путем в графе называется такая последовательность ребер, что конец i -го совпадает с началом $i+1$ -го. Длиной пути называется суммарный вес ребер. Путь является кратчайшим, если его длина минимальна.

Problem F. Дейкстра

Input file: *standard input*
Output file: *standard output*
Time limit: 2 second(s)
Memory limit: 512 MiB

Дан ориентированный взвешенный граф.

Найдите кратчайшее расстояние от одной заданной вершины до другой.

Input

В первой строке входного файла три числа: N , S и F ($1 \leq N \leq 2000, 1 \leq S, F \leq N$), где N — количество вершин графа, S — начальная вершина, а F — конечная. В следующих N строках по N чисел — матрица смежности графа, где -1 означает отсутствие ребра между вершинами, а любое целое неотрицательное число, не превосходящее 10 000 — присутствие ребра данного веса. На главной диагонали матрицы всегда нули.

Output

Вывести искомое расстояние или -1 , если пути не существует.

Example

standard input	standard output
3 1 2 0 -1 2 3 0 -1 -1 4 0	6

Problem G. Расстояние между вершинами - 2

Input file: *standard input*
Output file: *standard output*
Time limit: 2 second(s)
Memory limit: 512 MiB

Дан неориентированный взвешенный граф без петель и кратных рёбер. Найти вес минимального пути между двумя вершинами.

Input

Первая строка входного файла содержит натуральные числа N , M , вторая строка содержит натуральные числа S и F ($N \leq 5\,000$, $M \leq 100\,000$, $1 \leq S, F \leq N$, $S \neq F$) — количество вершин и рёбер графа а также номера вершин, длину пути между которыми требуется найти.

Следующие M строк по три натуральных числа b_i , e_i и w_i — номера концов i -ого ребра и его вес соответственно ($1 \leq b_i, e_i \leq n$, $0 \leq w_i \leq 100\,000$).

Output

Первая строка должна содержать одно натуральное число — вес минимального пути между вершинами S и F . Во второй строке через пробел выведите вершины на кратчайшем пути из S в F в порядке обхода.

Если путь из S в F не существует, выведите -1 .

Examples

standard input	standard output
4 4 1 3 1 2 1 2 3 2 3 4 5 4 1 4	3 1 2 3

Problem H. Бензоколонки

Input file: *standard input*
Output file: *standard output*
Time limit: 1 секунда
Memory limit: 512 мегабайт

В стране N городов, некоторые из которых соединены между собой дорогами. Для того, чтобы проехать по одной дороге, требуется один бак бензина. В каждом городе бак бензина имеет разную стоимость. Вам требуется добраться из первого города в N -ый, потратив как можно меньшее количество денег.

Input

В первой строке вводится число N ($1 \leq N \leq 100$), в следующей идет N чисел, i -ое из которых задает стоимость бензина в i -ом городе (все числа целые из диапазона от 0 до 100). Затем идет число M — количество дорог в стране, далее идет описание самих дорог. Каждая дорога задается двумя числами — номерами городов, которые она соединяет. Все дороги двусторонние (то есть по ним можно ездить как в одну, так и в другую сторону); между двумя городами всегда существует не более одной дороги; не существует дорог, ведущих из города в себя.

Output

Требуется вывести одно число — суммарную стоимость маршрута или -1, если добраться невозможно.

Examples

standard input	standard output
5 3 6 1 7 6 8 1 2 5 4 5 1 3 4 5 2 2 4 2 3 3 1	3

Problem I. Автобусы

Input file: *standard input*
Output file: *standard output*
Time limit: 1 секунда
Memory limit: 512 мебибайт

Между некоторыми деревнями края Васюки ходят автобусы. Поскольку пассажиропотоки здесь не очень большие, то автобусы ходят всего несколько раз в день.

Марии Ивановне требуется добраться из деревни d в деревню v как можно быстрее (считается, что в момент времени 0 она находится в деревне d).

Input

Сначала вводится число N — общее число деревень ($1 \leq N \leq 100$), затем номера деревень d и v , за ними следует количество автобусных рейсов R ($0 \leq R \leq 10^4$). Далее идут описания автобусных рейсов. Каждый рейс задается номером деревни отправления, временем отправления, деревней назначения и временем прибытия (все времена — целые от 0 до 10^4). Если в момент t пассажир приезжает в какую-то деревню, то уехать из нее он может в любой момент времени, начиная с t .

Output

Выведите минимальное время, когда Мария Ивановна может оказаться в деревне v . Если она не сможет с помощью указанных автобусных рейсов добраться из d в v , выведите -1 .

Examples

standard input	standard output
3 1 3 4 1 0 2 5 1 1 2 3 2 3 3 5 1 1 3 10	5

Problem J. Транспортировка кружек

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second(s)
Memory limit: 512 MiB

К Междупланетному Шахматному Конгрессу, который, как известно, проходит в городе Нью-Васюки, было решено подготовить памятные кружки для всех участников и болельщиков.

Имея привычку делать важные дела в самый последний момент, дизайнер закончил работу над макетом за два дня до начала школы. Ещё день уйдёт у завода-изготовителя на то, чтобы изготовить кружки и нанести на них изображение. На то, чтобы довести кружки от завода-изготовителя до Нью-Васюков, остаётся всего 24 часа.

Заказ на 10^7 экземпляров кружек (а именно столько заказали организаторы), конечно же, за один рейс не увезти. Однако, за первый рейс хочется привезти максимальное количество кружек. Для перевозки был заказан один большегрузный автомобиль. Но есть один нюанс: на некоторых дорогах установлено ограничение на вес автомобиля. Поэтому если автомобиль нагрузить кружками под завязку, то, возможно, не удастся воспользоваться самым коротким маршрутом, а придётся ехать в объезд. Может случиться даже так, что из-за этого грузовик не успеет доехать до Нью-Васюков вовремя, а этого допустить никак нельзя.

Итак, сколько же кружек можно погрузить в автомобиль, чтобы успеть привезти этот ценный груз вовремя, не нарушая правил дорожного движения?

Input

В первой строке находятся числа n ($1 \leq n \leq 500$) и m — количество узловых пунктов дорожной схемы и количество дорог, соответственно. В следующих m строках находится информация о дорогах. Каждая дорога описывается в отдельной строке следующим образом.

Сначала указаны номера узловых пунктов, которые соединяются данной дорогой, потом время, которое тратится на проезд по этой дороге, и, наконец, максимальный вес автомобиля, которому разрешено ехать по этой дороге. Известно, что все дороги соединяют различные пункты, причем для каждой пары пунктов есть не более одной дороги, непосредственно их соединяющей. Все числа разделены одним или несколькими пробелами.

Узловые пункты нумеруются числами от 1 до n . При этом завод по производству кружек имеет номер 1, а Нью-Васюки — номер n . Время проезда по дороге задано в минутах и не превосходит 1440 (24 часа). Ограничение на массу задано в граммах и не превосходит одного миллиарда. Кроме того, известно, что одна кружка весит 100 грамм, а пустой грузовик — 3 тонны.

Output

Выведите одно число - максимальное количество кружек, которое можно привезти за первый рейс, потратив не более 24 часов.

Examples

standard input	standard output
3 3 1 2 10 3000220 2 3 20 3000201 1 3 1 3000099	2
1 0	10000000

Problem A. Сумма простая

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second(s)
Memory limit: 512 MiB

Вам нужно научиться отвечать на запрос “сумма чисел на отрезке”.

Массив не меняется. Запросов много. Отвечать на 1 запрос следует за $O(1)$.

Input

Размер массива — n и числа x, y, a_0 , порождающие массив a : $a_i = (x \cdot a_{i-1} + y) \bmod 2^{16}$. Далее следуют количество запросов m и числа z, t, b_0 , порождающие массив b : $b_i = (z \cdot b_{i-1} + t) \bmod 2^{30}$, $c_i = b_i \bmod n$. i -й запрос — найти сумму на отрезке от $\min(c_{2i}, c_{2i+1})$ до $\max(c_{2i}, c_{2i+1})$ в массиве a .

Ограничения: $1 \leq n \leq 10^7, 0 \leq m \leq 10^7$. Все числа целые от 0 до 2^{16} . t может быть -1 .

Output

Выведите сумму всех сумм.

Example

standard input	standard output
3 1 2 3 3 1 -1 4	23

Note

$a = \{3, 5, 7\}, b = \{4, 3, 2, 1, 0, 2^{30} - 1\}, c = \{1, 0, 2, 1, 0, 0\}$,

запросы = $\{[0, 1], [1, 2], [0, 0]\}$, суммы = $\{8, 12, 3\}$.

Заметим, что вместо того, чтобы брать по модулю 2^{30} , достаточно всё считать в типе `int` и оставлять младшие 30 бит.

В Java есть знаковый сдвиг `>>` и беззнаковый `>>>`.

Одна из стандартных ошибок — переполнение типа. Проверьте, что у вас везде `int64`.

Problem B. Это моё болото!

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 512 MiB

В далеком царстве Байтландии есть болото, где живет огр Керш. Болото представляет из себя прямоугольное клеточное поле, где на некоторых клетках есть кочки. Керш очень любит гулять по своему болоту и готов делать это вечно!

Каждый день Керш планирует прогулку между двумя клетками в болоте. Известно, что стартовая клетка всегда находится не ниже и не правее, чем конечная клетка. Также, у Керша есть три условия на свою прогулку.

- Все клетки пути обязаны быть кочками
- Керш может ходить только вниз на одну клетку или вправо на одну клетку
- Путь должен представлять из себя ломаную с не более чем двумя изгибами.

Ясно, что есть пары клеток в болоте, между которыми нельзя прогуляться, соблюдая все правила. Помогите Кершу для каждого своего маршрута узнать сможет ли он его совершить.

Input

В первой строке вводятся два числа N, M - размеры поля. ($1 \leq N \times M \leq 10^6$).

В следующих N строках вводятся строки длиной M из нулей и единиц. Единица в i -той строке на j -том месте означает, что в клетке болота с координатами (i, j) есть кочка, ноль обозначает, что кочки нет.

Далее вводится число Q - количество прогулок Керша по болоту. ($1 \leq Q \leq 10^6$).

В следующих Q строках вводятся данные об очередной прогулке в формате $x_1 y_1 x_2 y_2$ ($1 \leq x_1 \leq x_2 \leq N, 1 \leq y_1 \leq y_2 \leq M$). Данная строка обозначает, что Керш хочет начать свою прогулку в клетке с координатами (x_1, y_1) , а закончить хочет в клетке с координатами (x_2, y_2) .

Output

Выведите в одну строку Q чисел, где i -тое число равно единице, если i -тую прогулку можно совершить, соблюдая все правила, либо нулю, если это сделать нельзя.

Example

standard input	standard output
6 6 111001 101001 101111 010010 010011 011100 5 1 1 3 6 1 2 4 5 4 2 4 2 1 6 5 6 2 4 3 4	1 0 1 0 0

Problem C. Линейная сумма

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second(s)
Memory limit: 512 MiB

Есть n случайных точек на прямой с координатами от 0 до $2^{32}-1$. У каждой точки есть значение от 0 до $2^{32}-1$. Вам нужно обработать q случайных запросов вида “сумма значений точек, с координатами от l до r включительно”.

Input

На первой строке числа n, q . ($1 \leq n \leq 2^{20}, 1 \leq q \leq 2^{23}$). На второй строке пара целых чисел a, b от 1 до 10^9 , используемая в генераторе случайных чисел.

```
1. unsigned int cur = 0; // беззнаковое 32-битное число
2. unsigned int nextRand24() {
3.     cur = cur * a + b; // вычисляется с переполнениями
4.     return cur >> 8; // число от 0 до  $2^{24}-1$ .
5. }
6. unsigned int nextRand32() {
7.     unsigned int a = nextRand24(), b = nextRand24();
8.     return (a << 8) ^ b; // число от 0 до  $2^{32}-1$ .
9. }
```

Каждая точка генерируется следующим образом:

```
1. value = nextRand32(); // значение точки
2. x = nextRand32(); // координата точки
```

Каждый запрос генерируется следующим образом:

```
1. l = nextRand32();
2. r = nextRand32();
3. if (l > r) swap(l, r); // получили отрезок [l..r]
```

Сперва генерируются точки, затем запросы.

Output

Выведите сумму ответов на все запросы по модулю 2^{32} .

Examples

standard input	standard output
5 5 13 239	3950632748

Note

```
p = {value, x}
p[0] = {13, 41645}
p[1] = {7695587, 1253435649}
p[2] = {749170640, 2683600557}
p[3] = {2444595881, 1270561959}
p[4] = {3436107648, 486388002}
```

Problem D. Хороша ли подстрока?

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

Вам дана строка из круглых скобок. Нужно отвечать на запросы вида “есть ли у подстроки $[l..r]$ исходной строки циклический сдвиг, являющийся правильной скобочной последовательностью (ПСП)?”.

Напомним определение ПСП:

- Пустая строка – ПСП.
- Если s – ПСП, то $'(s)'$ – тоже.
- Если s_1, s_2 – ПСП, то $'s_1s_2'$ – тоже.

Input

Строка s из круглых скобок длины от 1 до 10^6 .

На второй строке количество запросов q ($1 \leq q \leq 10^5$).

Следующие q строк содержат пары чисел l, r ($1 \leq l \leq r \leq |s|$).

Output

Для каждого запроса, если ответ да, выведите 1, иначе 0.

Examples

standard input	standard output
<pre>()()(6 1 2 1 4 5 6 3 6 1 5 2 5</pre>	111100

Problem E. RMQ

Input file: *standard input*
Output file: *standard output*
Time limit: 2 секунды
Memory limit: 512 MiB

Реализуйте структуру данных, которая на данном массиве из N целых чисел позволяет узнать максимальное значение на этом массиве и индекс элемента, на котором достигается это максимальное значение.

Input

В первой строке вводится натуральное число N ($1 \leq N \leq 10^5$) – количество элементов в массиве. В следующей строке содержатся N целых чисел, не превосходящих по модулю 10^9 – элементы массива. Далее идет число K ($0 \leq K \leq 10^5$) – количество запросов к структуре данных. Каждая из следующих K строк содержит два целых числа l и r ($1 \leq l \leq r \leq N$) – левую и правую границы отрезка в массиве для данного запроса.

Output

Для каждого из запросов выведите два числа: наибольшее значение среди элементов массива на отрезке от l до r и индекс одного из элементов массива, принадлежащий отрезку от l до r , на котором достигается этот максимум.

Examples

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

Problem F. Сумма

Input file: *standard input*
Output file: *standard output*
Time limit: 2 second(s)
Memory limit: 512 MiB

Дан массив из N элементов, нужно научиться находить сумму чисел на отрезке.

Input

Первая строка содержит два целых числа N и K — число чисел в массиве и количество запросов. ($1 \leq N \leq 100\,000$), ($0 \leq K \leq 100\,000$). Следующие K строк содержат запросы

- “A i x ” — присвоить i -му элементу массива значение x ($1 \leq i \leq n$, $0 \leq x \leq 10^9$)
- “Q l r ” — найти сумму чисел в массиве на позициях от l до r . ($1 \leq l \leq r \leq n$)

Изначально в массиве живут нули.

Output

На каждый запрос вида Q l r нужно вывести единственное число — сумму на отрезке.

Examples

standard input	standard output
5 9	0
A 2 2	2
A 3 1	1
A 4 2	2
Q 1 1	0
Q 2 2	5
Q 3 3	
Q 4 4	
Q 5 5	
Q 1 5	

Problem G. Range Variation Query

Input file: *standard input*
Output file: *standard output*
Time limit: 2 second(s)
Memory limit: 512 MiB

В начальный момент времени последовательность a_n задана следующей формулой: $a_n = n^2 \bmod 12345 + n^3 \bmod 23456$.

Требуется много раз отвечать на запросы следующего вида:

- найти разность между максимальным и минимальным значениями среди элементов a_i, a_{i+1}, \dots, a_j ;
- присвоить элементу a_i значение j .

Input

Первая строка входного файла содержит натуральное число k — количество запросов ($1 \leq k \leq 100\,000$). Следующие k строк содержат запросы, по одному на строке. Запрос номер i описывается двумя целыми числами x_i, y_i .

Если $x_i > 0$, то требуется найти разность между максимальным и минимальным значениями среди элементов a_{x_i}, \dots, a_{y_i} . При этом $1 \leq x_i \leq y_i \leq 100\,000$.

Если $x_i < 0$, то требуется присвоить элементу $a_{|x_i|}$ значение y_i . В этом случае $-100\,000 \leq x_i \leq -1$ и $|y_i| \leq 100\,000$.

Output

Для каждого запроса первого типа в выходной файл требуется вывести одну строку, содержащую разность между максимальным и минимальным значениями на соответствующем отрезке.

Examples

standard input	standard output
7	34
1 3	68
2 4	250
-2 -100	234
1 5	1
8 9	
-3 -101	
2 3	

Problem H. Знакочередование

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second(s)
Memory limit: 512 MiB

Реализуйте структуру данных из n элементов $a_1, a_2 \dots a_n$, поддерживающую следующие операции:

- присвоить элементу a_i значение j ;
- найти знакопеременную сумму на отрезке от l до r включительно $(a_l - a_{l+1} + a_{l+2} - \dots \pm a_r)$.

Input

В первой строке входного файла содержится натуральное число n ($1 \leq n \leq 10^5$) — длина массива. Во второй строке записаны начальные значения элементов (неотрицательные целые числа, не превосходящие 10^4).

В третьей строке находится натуральное число m ($1 \leq m \leq 10^5$) — количество операций. В последующих m строках записаны операции:

- операция первого типа задается тремя числами $0 \ i \ j$ ($1 \leq i \leq n, 1 \leq j \leq 10^4$).
- операция второго типа задается тремя числами $1 \ l \ r$ ($1 \leq l \leq r \leq n$).

Output

Для каждой операции второго типа выведите на отдельной строке соответствующую знакопеременную сумму.

Examples

standard input	standard output
3	-1
1 2 3	2
5	-1
1 1 2	3
1 1 3	
1 2 3	
0 2 1	
1 1 3	

Problem I. Количество инверсий

Input file: *standard input*
Output file: *standard output*
Time limit: 2 second(s)
Memory limit: 512 MiB

Дан массив случайных целых чисел, нужно найти количество инверсий.

Input

На первой строке числа n ($1 \leq n \leq 1\,000\,000$) — размер массива и m ($1 \leq m \leq 2^{24}$) числа в массиве от 0 до $m - 1$). На второй строке пара целых чисел a, b от 1 до 10^9 , используемая в генераторе случайных чисел.

```
1. unsigned int cur = 0; // беззнаковое 32-битное число
2. unsigned int nextRand24() {
3.     cur = cur * a + b; // вычисляется с переполнениями
4.     return cur >> 8; // число от 0 до  $2^{24} - 1$ .
5. }
```

Элементы массива генерируются последовательно. $x_i = \text{nextRand24}() \% m$;

Output

Выведите количество инверсий.

Examples

standard input	standard output
20 5 19 18	63

Note

Сгенерированный массив: 01142210424031343330.