Задача A. Два числа

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

Даны два целых числа A и B ($1 \leqslant A, B \leqslant 100$). Найдите два таких целых числа X и Y, что выполнено равенство $AX + BY = 1$.

Формат входных данных

Во первой строке записаны два числа A и B, разделённые пробелом.

Формат выходных данных

В первой строке выведите два числа X и Y, разделённые пробелом. Требуется, чтобы выполнялись неравенства $|X| \leqslant 10\,000$ и $|Y| \leqslant 10\,000$. Если правильных ответов несколько, разрешается вывести любой из них. Если же таких чисел не существует, выведите вместо них два нуля.

Примеры

стандартный ввод	***стандартный вывод***
`2 3`	`2 -1`
`4 6`	`0 0`
`100 51`	`-5075 9951`

Задача B. Целые точки

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

Точку на координатной плоскости будем называть целой, если обе её координаты — целые числа. К примеру, точки $(0, 0)$ и $(-4, 7)$ — целые, а точки $(-1, 0.5)$ и $(\frac{1}{3}, \sqrt{2})$ — нет.

Сколько целых точек содержит заданный отрезок на плоскости?

Формат входных данных

В первой строке заданы два числа x_1 и y_1 — координаты одного конца отрезка. Во второй строке заданы два числа x_2 и y_2 — координаты другого конца отрезка. Числа в каждой строке разделены пробелами. Все заданные координаты — целые числа, не превосходящие по модулю $1\,000\,000\,000$. Гарантируется, что заданные две точки не совпадают.

Формат выходных данных

Выведите количество целых точек на заданном отрезке. Обратите внимание, что концы отрезка тоже учитываются.

Примеры

стандартный ввод	***стандартный вывод***
`2 1` `4 1`	`3`
`0 0` `5 7`	`2`

Пояснения к примерам

В первом примере целые точки — $(2, 1)$, $(3, 1)$ и $(4, 1)$.

Во втором примере целые точки — только концы отрезка $(0, 0)$ и $(5, 7)$.

Задача C. Волшебные ночи

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

Вокруг далёкой планеты Этан вращается три луны: Клементина, Лея и Матильда. Каждую k-ю ночь наступает полнолуние Клементины, каждую l-ю ночь — полнолуние Леи, а каждую m-ю ночь — полнолуние Матильды. В году на этой планете n ночей, а Новый Год наступает днём.

Ночь на планете Этан считается волшебной, если в эту ночь наступает полнолуние хотя бы у одной из лун. Известно, что в последнюю ночь прошлого года полнолуние наступило одновременно у всех трёх лун Этана. Сколько волшебных ночей в текущем году?

Формат входных данных

В первой строке заданы четыре целых числа k, l, m и n ($1 \leqslant k, l, m, n \leqslant 10^9$). Числа разделены пробелами.

Формат выходных данных

В первой строке выведите одно целое число — количество волшебных ночей в текущем году.

Примеры

стандартный ввод	*стандартный вывод*
3 4 5 10	7
5 5 5 10	2
30 29 31 360	35
2 4 6 5	2

Пояснения к примерам

В первом примере волшебными считаются 3-я, 4-я, 5-я, 6-я, 8-я, 9-я и 10-я ночи.

Во втором примере волшебных ночей только две — 5-я и 10-я ночи.

В третьем примере волшебными оказываются 12 ночей, когда полнолуние наступает у Клементины, 12 ночей, когда полнолуние наступает у Леи, и 11 ночей, когда полнолуние наступает у Матильды.

В четвёртом примере во вторую ночь наступает полнолуние Клементины, а в четвёртую — Клементины и Леи. Поскольку в году всего пять ночей, следующее полнолуние Матильды случится только в следующем году.

Задача D. Делители 1

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

По данному числу N определите количество его различных положительных делителей.

Формат входных данных

В первой строке задано единственное целое число N ($1 \leqslant N \leqslant 10^{15}$).

Формат выходных данных

Выведите единственное число k — количество различных положительных делителей числа N.

Примеры

стандартный ввод	*стандартный вывод*
1	1
2	2
6	4
29	2
48	10

В последнем примере число 48 имеет десять делителей — это числа 1, 2, 3, 4, 6, 8, 12, 16, 24 и 48.

Задача E. Обратный элемент лайт

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

Дано простое число n. Рассмотрим множество Z_n, элементами которого являются целые числа $0, 1, 2, \ldots, n-1$. Элемент q называется *обратным* к элементу p, если $(p \cdot q) \bmod n = 1$.

Найдите обратный элемент q по заданным числам n и p.

Формат входных данных

В первой строке заданы два целых числа n и p через пробел ($2 \leqslant n \leqslant 10^9$, число n простое, $0 \leqslant p < n$).

Формат выходных данных

Если у числа p нет в множестве Z_n обратного элемента, выведите -1.

В противном случае выведите q.

Примеры

стандартный ввод	***стандартный вывод***
3 2	2
5 2	3
7 4	2
17 0	-1

Задача F. Обратный элемент

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

Дано целое число $n > 0$. Рассмотрим множество Z_n, элементами которого являются целые числа $0, 1, 2, \ldots, n-1$. Элемент q называется *обратным* к элементу p, если $(p \cdot q) \bmod n = 1$.

Найдите обратный элемент q по заданным числам n и p.

Формат входных данных

В первой строке заданы два целых числа n и p через пробел ($1 \leqslant n \leqslant 10^9$, $0 \leqslant p < n$).

Формат выходных данных

Если у числа p нет в множестве Z_n обратного элемента, выведите -1.

В противном случае выведите q.

Примеры

стандартный ввод	***стандартный вывод***
3 2	2
5 2	3
8 4	-1
17 0	-1

Задача G. Первообразный корень

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

Мультипликативный порядок числа q по модулю m — это минимальное целое положительное число k, для которого $q^k \bmod m = 1$.

Число q называется *первообразным корнем* по простому модулю p, если мультипликативный порядок q равен $p - 1$.

Дано простое число p и набор чисел $q_1, q_2, \ldots, q_n$. Для каждого q_i выясните, является ли оно первообразным корнем по модулю p.

Формат входных данных

В первой строке заданы через пробел два целых числа n и p — количество чисел и модуль ($1 \leqslant n \leqslant 100$, $2 \leqslant p \leqslant 10^9$, число p является простым). Следующие n строк содержат по одному числу q_i каждая ($0 < q_i < p$).

Формат выходных данных

Выведите n строк; в i-й строке выведите «YES», если q_i является первообразным корнем по модулю p, и «NO» в противном случае.

Примеры

стандартный ввод	**стандартный вывод**
1 3 2	YES
2 7 2 3	NO YES

Задача H. Делители 2

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

Натуральное число a называется *делителем* натурального числа b, если $\frac{b}{a}$ — также натуральное число. Например, 1, 2, 3 и 6 — делители числа 6, а 4, 5 и 7 не являются его делителями.

В этой задаче требуется определить, каково максимальное количество различных делителей, которое может иметь натуральное число от 1 до N, включительно, и найти минимальное из чисел на этом интервале, имеющее ровно столько делителей.

Формат входных данных

В первой строке задано число N ($1 \leqslant N \leqslant 10^{18}$).

Формат выходных данных

Выведите два целых числа через пробел — сколько делителей может иметь натуральное число от 1 до N, включительно, а также само минимальное натуральное число, имеющее столько делителей.

Примеры

стандартный ввод	**стандартный вывод**
2	2 2
5	3 4
7	4 6
18	6 12

Задача I. Дискретное логарифмирование лайт

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

Даны натуральные числа a, b и n. Требуется найти *дискретный логарифм* b по основанию a по модулю n, то есть такое число x ($0 \leqslant x < n$), что $a^x \equiv b \pmod{n}$.

Формат входных данных

В первой строке заданы через пробел три целых числа a, b и n ($2 \leqslant n \leqslant 10^9$, число n простое, $0 < a, b < n$).

Формат выходных данных

В первой строке выведите -1, если дискретного логарифма не существует. Иначе следует вывести его значение.

Если ответ неоднозначен, разрешается выводить любой.

Пример

стандартный ввод	*стандартный вывод*
3 5 7	5

Задача J. Дискретное логарифмирование

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

Даны натуральные числа a, b и n. Требуется найти *дискретный логарифм* b по основанию a по модулю n, то есть такое число x ($0 \leqslant x < n$), что $a^x \equiv b \pmod{n}$.

Формат входных данных

В первой строке заданы через пробел три целых числа a, b и n ($1 \leqslant a, b, n \leqslant 10^{12}$).

Формат выходных данных

В первой строке выведите -1, если дискретного логарифма не существует. Иначе следует вывести его значение.

Если ответ неоднозначен, разрешается выводить любой.

Пример

стандартный ввод	*стандартный вывод*
2 4 6	2

Задача K. Проверка на простоту 1

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

Натуральное число называется *простым*, если оно делится нацело только на себя и на 1. Единица простым числом не считается.

Дано число p. Определите, простое ли оно.

Формат входных данных

В первой строке задано целое число p ($1 \leqslant p \leqslant 10^9$).

Формат выходных данных

В первой строке выведите «YES», если число простое, и «NO» в противном случае.

Примеры

стандартный ввод	***стандартный вывод***
3	YES
6	NO

Задача L. Проверка на простоту 2

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

Натуральное число называется *простым*, если оно делится нацело только на себя и на 1. Единица простым числом не считается.

Дано число p. Определите, простое ли оно.

Формат входных данных

В первой строке задано целое число p ($1 \leqslant p \leqslant 10^{18}$).

Формат выходных данных

В первой строке выведите «YES», если число простое, и «NO» в противном случае.

Примеры

стандартный ввод	***стандартный вывод***
3	YES
6	NO

Задача M. Тест Миллера–Рабина

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

Существует множество способов проверки числа на простоту. Например, если проверяемое число N достаточно мало, то можно просто поделить N на все простые числа, не превосходящие $\sqrt{N}$. Если N делится нацело хотя бы на одно из них — значит, оно составное, в противном же случае оно является простым.

Однако, когда число N велико, такой метод может потребовать от проверяющего слишком много времени — ведь трудоёмкость растёт экспоненциально от длины числа N. В настоящее время известно несколько способов определить простоту числа точно, но все они работают довольно долго.

Поэтому чаще применяют способы, определяющие простоту числа с некоторой вероятностью. Один из наиболее быстрых и вместе с тем довольно надёжных способов известен как тест Миллера–Рабина. Ознакомимся с ним подробнее.

Сначала проверим, что N нечётно и больше, чем 1 (в противном случае проверка тривиальна). Представим $N-1$ как $2^s \cdot d$; заметим, что $s \geqslant 1$.

Теперь для нескольких различных $a \in [1, N-1]$ произведём следующую процедуру. Рассмотрим числа $k_r = a^{2^r \cdot d}$ для $r = 0, 1, \ldots, s-1$. Если $k_0 \bmod N \neq 1$ и ни одно из k_r не совпадает с -1 по модулю N (другими словами, $k_r \bmod N \neq N-1$), число N — составное. В противном случае мы повторяем эту процедуру для следующего a. Чем больше чисел a было проверено, тем больше вероятность того, что число N — простое. Обычно в качестве a подставляют первые несколько простых чисел — $2, 3, 5, 7, 11, \ldots$.

Мы не будем сейчас останавливаться на том, почему тест Миллера–Рабина работает. Наша задача заключается в другом — по числу N определить, каково же наименьшее простое число a, для которого описанная выше процедура приведёт к установлению того, что N — составное (разумеется, если это так). Число a не окажется слишком большим — известно, что наименьшее нечётное составное число N такое, что для него не срабатывают проверки с $a = 2, 3, 5, 7, 11, 13, 17, 19$, равно $341\,550\,071\,728\,321$.

Формат входных данных

В первой строке записано число N ($1 \leqslant N \leqslant 10^{14}$).

Формат выходных данных

Если N чётно или равно единице, и тест Миллера–Рабина неприменим, выведите -1. Если число N нечётное и простое, выведите 0. Иначе выведите наименьшее такое простое число a, что при его проверке по приведённому выше алгоритму выяснится, что N — составное.

Примеры

стандартный ввод	**стандартный вывод**
2	-1
15	2
4	-1
821	0
2047	3

Задача N. Простые числа

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

Простым, как известно, называется натуральное число, которое делится нацело только на себя и на единицу. Число, делящееся на другое натуральное число, меньшее его, называется составным. Единица не считается ни простым, ни составным числом. Так, есть 25 простых чисел, не превосходящих 100 — это числа 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97.

В этой задаче мы попробуем выяснить, сколько простых чисел расположено на отрезке $[A, B]$, где A и B — целые и $A \leqslant B$. Математики интересовались подобными вопросами уже давно. Ещё в середине XIX века француз Джозеф Луи Франсуа Бертран выдвинул гипотезу о том, что для любого $n > 1$ между n и $2n$ есть по крайней мере одно простое число. Эта гипотеза была впоследствии доказана Пафнутием Львовичем Чебышёвым и получила название теоремы Чебышёва. Другая теорема, связывающая имена этих двух математиков, говорит о том, что количество простых чисел от 1 до n ведёт себя примерно как $\frac{n}{\ln n}$.

Возможности современных вычислительных машин позволяют посчитать количество простых чисел от A до B точно, если A и B достаточно невелики. В этом и состоит предлагаемая задача.

Формат входных данных

В первой строке записаны два числа — A и B $(1 \leqslant A \leqslant B \leqslant 50\,000\,000)$.

Формат выходных данных

Выведите одно число — количество простых чисел на отрезке $[A, B]$.

Примеры

стандартный ввод	***стандартный вывод***
`1 2`	`1`
`2 3`	`2`
`1 100`	`25`
`98 98`	`0`
`97 97`	`1`

Задача O. Фи

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

В этот раз ваша задача очень простая. Всего лишь сосчитайте сумму значений функции Эйлера от a до b включительно, то есть

$$\sum_{i=a}^{b} \varphi(i).$$

Здесь функция Эйлера $\varphi(n)$ — это количество целых чисел от 1 до n включительно, взаимно простых с n.

Формат входных данных

В первой строке заданы два целых числа a и b $(1 \leqslant a \leqslant b \leqslant 4 \cdot 10^{12}$, $b - a \leqslant 2 \cdot 10^6)$.

Формат выходных данных

Выведите значение суммы.

Пример

стандартный ввод	***стандартный вывод***
`2 4`	`5`

Пояснение к примеру

В примере $\varphi(2) + \varphi(3) + \varphi(4) = 1 + 2 + 2 = 5$.

Задача P. Произведение матриц

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

Произведением матриц A и B размера $p \times q$ и $q \times r$, соответственно, называется матрица C размера $p \times r$, элементы которой вычисляются по формуле

$$C_{i,j} = \sum_{k=1}^{q} A_{i,k} \cdot B_{k,j}.$$

По данным матрицам A и B найдите их произведение.

Формат входных данных

В первой строке заданы через пробел три целых числа p, q и r ($1 \leqslant p, q, r \leqslant 100$). В следующих p строках записана матрица A; каждая из этих строк содержит q целых чисел, разделённых пробелами. Наконец, в последних q строках записана матрица B; каждая из этих строк содержит r целых чисел, разделённых пробелами. Элементы матриц не превосходят 100 по абсолютной величине.

Формат выходных данных

Выведите матрицу C — p строк, в каждой из которых — r чисел через пробел.

Примеры

стандартный ввод	***стандартный вывод***
2 2 2 1 0 0 1 1 0 0 1	1 0 0 1
1 3 1 1 2 3 -1 -2 -3	-14
3 2 4 0 1 1 0 0 1 2 1 0 0 1 1 2 1	1 1 2 1 2 1 0 0 1 1 2 1

Задача Q. Числа Фибоначчи по модулю 1

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

Числа Фибоначчи F_0, F_1, F_2, $\ldots$, F_n определяются следующим образом: $F_0 = 0$, $F_1 = 1$, а для любого $n > 1$ выполнено равенство $F_n = F_{n-1} + F_{n-2}$.

По заданному числу n выведите остаток от деления числа Фибоначчи F_n на m.

Формат входных данных

В первой строке ввода заданы через пробел два целых числа n и m ($0 \leqslant n \leqslant 10^9$, $1 \leqslant m \leqslant 10^9$).

Формат выходных данных

В первой строке выведите одно число: остаток от деления F_n на m.

Пример

стандартный ввод	**стандартный вывод**
`8 12345`	`21`
`100 1000000000`	`261915075`

Пояснение к примеру

Первые несколько чисел Фибоначчи таковы: $F_0 = 0$, $F_1 = 1$, $F_2 = 1$, $F_3 = 2$, $F_4 = 3$, $F_5 = 5$, $F_6 = 8$, $F_7 = 13$, $F_8 = 21$.

Число Фибоначчи с номером 100 равно 354 224 848 179 261 915 075. Остаток от деления этого числа на 1 000 000 000 — это последние девять цифр его десятичной записи.

Задача R. Числа Фибоначчи по модулю 2

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

Числа Фибоначчи F_0, F_1, F_2, $\ldots$, F_n определяются следующим образом: $F_0 = 0$, $F_1 = 1$, а для любого $n > 1$ выполнено равенство $F_n = F_{n-1} + F_{n-2}$.

По заданному числу n выведите остаток от деления числа Фибоначчи F_n на m.

Формат входных данных

В первой строке ввода заданы через пробел два целых числа n и m ($0 \leqslant n \leqslant 10^{18}$, $1 \leqslant m \leqslant 10^{18}$).

Формат выходных данных

В первой строке выведите одно число: остаток от деления F_n на m.

Пример

стандартный ввод	**стандартный вывод**
`8 12345`	`21`
`100 1000000000`	`261915075`

Пояснение к примеру

Первые несколько чисел Фибоначчи таковы: $F_0 = 0$, $F_1 = 1$, $F_2 = 1$, $F_3 = 2$, $F_4 = 3$, $F_5 = 5$, $F_6 = 8$, $F_7 = 13$, $F_8 = 21$.

Число Фибоначчи с номером 100 равно 354 224 848 179 261 915 075. Остаток от деления этого числа на 1 000 000 000 — это последние девять цифр его десятичной записи.

Задача S. Зеркальный лабиринт

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

В зеркальном лабиринте живут солнечные зверьки: зайчики, кролики и тушканчики. Пока свет в лабиринте погашен, в лабиринте нет никаких солнечных зверьков. Как только свет включают, каждую секунду происходят превращения. В k-ю секунду с момента включения одновременно происходят следующие преобразования:

- В лабиринте появляется k новых солнечных зайчиков.
- Каждый солнечный зайчик, появившийся на предыдущей секунде, исчезает, превращаясь в a кроликов.
- Каждый солнечный кролик, появившийся на предыдущей секунде, исчезает, разделяясь на несколько частей: одного зайчика, b кроликов и одного тушканчика.
- Каждый солнечный тушканчик, появившийся на предыдущей секунде, исчезает, производя на свет c зайчиков и трёх кроликов.

Кроме того, лабиринт вмещает ограниченное количество зверьков каждого типа. Поэтому в конце каждой секунды — то есть после всех превращений, произошедших в течение этой секунды — если количество s зверьков какого-либо из трёх типов больше или равно m, зверьков этого типа остаётся $s \bmod m$ (остаток от целочисленного деления s на m).

По заданным числам n, m, a, b и c найдите, сколько зверьков каждого из трёх типов в отдельности окажется в лабиринте после того, как свет будет включён в течение n секунд.

Формат входных данных

В первой строке ввода заданы через пробел пять целых чисел: n, m, a, b и c ($1 \leqslant n \leqslant 10^9$, $1 \leqslant m \leqslant 10^9$, $0 \leqslant a, b, c < m$).

Формат выходных данных

В первой строке выведите три числа: количество солнечных зайчиков, солнечных кроликов и солнечных тушканчиков в лабиринте после того, как свет будет включён в течение n секунд. Разделяйте соседние числа в строке пробелами.

Примеры

стандартный ввод	стандартный вывод
1 10 2 3 4	1 0 0
2 10 2 3 4	2 2 0
3 10 2 3 4	5 0 2
4 10 2 3 4	2 6 0

Пояснения к примерам

В примерах представлены первые четыре секунды преобразований при $a = 2$, $b = 3$, $c = 4$ и $m = 10$.

На первой секунде появляется один солнечный зайчик.

На второй секунде он превращается в двух солнечных кроликов, а также появляются два новых солнечных зайчика.

На третьей секунде зайчики, исчезая, порождают четырёх новых кроликов, а кролики — двух зайчиков, шестерых кроликов и двух тушканчиков. Заметим, что количество солнечных кроликов становится равным $a \cdot 2 + b \cdot 2 = 2 \cdot 2 + 3 \cdot 2 = 10$. Поэтому в конце третьей секунды их численность падает до $10 \bmod 10 = 0$. Кроме того, появляется ещё три новых солнечных зайчика.

На четвёртой секунде появляется четыре новых солнечных зайчика, пять старых зайчиков, исчезая, порождают десять кроликов, а два старых тушканчика — восьмерых зайчиков и шестерых кроликов. Поскольку $m = 10$, из $4 + 8 = 12$ зайчиков и $10 + 6 = 16$ кроликов остаётся $12 \bmod 10 = 2$ зайчика и $16 \bmod 10 = 6$ кроликов.

Задача T. Школьницы

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	5 секунд
Ограничение по памяти:	512 мебибайт

Школьница Алиса на уроке технологии познакомилась с шарнирными механизмами. Она сконструировала инструмент, позволяющий по трём вершинам параллелограмма (возможно, вырожденного) достраивать четвёртую. Формально, по трём точкам A, B, C (из которых некоторые или все могут совпадать или лежать на одной прямой) она умеет строить такую точку D, что векторы $\overrightarrow{AB}$ и $\overrightarrow{DC}$ равны.

Школьница Алина на уроке геометрии познакомилась с понятием правильного многоугольника. В этой задаче мы будем пользоваться следующими определениями:

- будем говорить, что точки $A_1, A_2, \ldots, A_n$ ($n \geqslant 3$) *образуют вырожденный правильный многоугольник*, если все эти точки совпадают;
- будем говорить, что точки $A_1, A_2, \ldots, A_n$ ($n \geqslant 3$) *образуют невырожденный правильный многоугольник против часовой стрелки*, если они все попарно различны, лежат на одной окружности с некоторым центром O, и $\angle A_1OA_2 = \angle A_2OA_3 = \ldots = \angle A_nOA_1 = \frac{360^\circ}{n} = \frac{2\pi}{n}$, причём во всех этих углах поворот с центром в O на $\frac{2\pi}{n}$ против часовой стрелки переводит $\overrightarrow{OA_i}$ в $\overrightarrow{OA_{(i \bmod n)+1}}$;
- будем говорить, что точки $A_1, A_2, \ldots, A_n$ ($n \geqslant 3$) *образуют невырожденный правильный многоугольник*, если существует их перестановка $A_{(1)}, A_{(2)}, \ldots, A_{(n)}$, образующая невырожденный правильный многоугольник против часовой стрелки;
- будем говорить, что точки $A_1, A_2, \ldots, A_n$ ($n \geqslant 3$) *образуют правильный многоугольник*, если они образуют вырожденный правильный многоугольник или образуют невырожденный правильный многоугольник.

Обратите внимание, что последнее определение не зависит от порядка точек: если список точек образует правильный многоугольник, то любая их перестановка тоже образует правильный многоугольник.

Завуч Арина решила проверить умения школьниц. Сначала она дала им задание — построить на плоскости $n+m$ точек. Первые n точек должны образовывать невырожденный правильный многоугольник против часовой стрелки. Каждая из следующих m точек строится по каким-то трём предыдущим точкам с помощью инструмента Алисы.

Девочки справились с этой частью задания. Затем Арина стала называть некоторые наборы точек и спрашивать: образуют ли они правильный многоугольник? Это уже оказалось весьма трудным для школьниц, поэтому они обратились за помощью к вам. Напишите программу, которая справится с Арининой задачей.

Формат входных данных

В первой строке заданы три целых числа n, m, k — число вершин в исходном правильном многоугольнике, число точек, дополнительно построенных при помощи Алисиного инструмента, и число многоугольников, про которые спросит Арина ($3 \leqslant n \leqslant 10^4$, $0 \leqslant m \leqslant 3 \cdot 10^4$, $1 \leqslant k \leqslant 10^4$). Точки $K_1, K_2, \ldots, K_n$ образуют невырожденный правильный многоугольник против часовой стрелки.

В следующих m строках описано, как строятся точки $K_{n+1}, \ldots, K_{n+m}$. В i-й строке заданы три целых числа a_i, b_i, c_i ($1 \leqslant a_i, b_i, c_i \leqslant n+i-1$) — номера трёх точек, к которым применяется инструмент Алисы. Точка K_{n+i} определяется так, что $\overrightarrow{K_{a_i}K_{b_i}} = \overrightarrow{K_{n+i}K_{c_i}}$. Некоторые или все из чисел a_i, b_i, c_i могут совпадать.

В следующих k строках описаны наборы точек Арины. В i-й строке описывается i-й набор в формате «r_i $P_1^{(i)}$ $P_2^{(i)}$ $\ldots$ $P_{r_i}^{(i)}$». Это значит, что школьницам требуется проверить, что точки $K_{P_1^{(i)}}, \ldots, K_{P_{r_i}^{(i)}}$ образуют правильный многоугольник ($3 \leqslant r_i \leqslant 3 \cdot 10^4$, $1 \leqslant P_j^{(i)} \leqslant n+m$). Гарантируется, что сумма всех r_i не превосходит $3 \cdot 10^4$. Некоторые или все из чисел $P_j^{(i)}$ могут совпадать.

Формат выходных данных

Выведите k строк В i-й строке должно быть слово «`Yes`», если i-й набор Арины образует правильный многоугольник, и «`No`» иначе. Каждая из букв вывода может быть в любом регистре (прописной или строчной).

Пример

стандартный ввод	*стандартный вывод*
3 6 8	Yes
1 2 3	Yes
3 1 4	Yes
5 4 3	No
3 1 2	No
4 5 3	No
4 5 2	Yes
6 4 7 6 5 1 2	No
3 1 3 2	
3 1 1 8	
4 2 5 6 7	
3 2 1 4	
3 6 5 9	
3 4 7 9	
4 1 3 2 8	

Замечание

Иллюстрация к примеру: