

## Problem A. Адвокат

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

Байтазар — известный адвокат, совладелец компании «Байтазар и товарищи». Также он участвует в заседаниях различных комитетов и комиссий. Неудивительно, что он всё время занят.

Каждый день Байтазара приглашают на множество различных заседаний; в какой-то момент он понял, что ему сложно понять, сможет ли он участвовать во всех заседаниях.

Поэтому Байтазар нанял секретаря, который должен решать эти вопросы. Байтазар решил, что каждый день он будет участвовать только в двух заседаниях, но с начала и до конца. На оставшиеся заседания Байтазар отправит своих помощников.

Считается, что заседания не перекрываются, если одно из них начинается строго после того, как завершится другое. Помогите секретарю Байтазара составить соответствующее расписание.

### Input

Первая строка входного файла содержит два целых числа  $n$  и  $m$  ( $2 \leq n \leq 500\,000$ ,  $1 \leq m \leq 20$ ) — количество совещаний, на которые приглашён Байтазар и количество дней в его расписании. describing the number of meetings in Byteasar's schedule and the number of days included in it.

Каждая из последующих  $n$  строк описывает одну встречу. Описание встречи состоит из трёх целых чисел  $a_i, b_i, d_i$  ( $1 \leq a_i < b_i \leq 80\,000\,000$ ,  $1 \leq d_i \leq m$ ), обозначающих, что в день  $d_i$  Байтазар приглашён на встречу, которая начинается на  $a_i$  миллисекунде соответствующего дня и завершается на  $b_i$  миллисекунде.

### Output

Выведите  $m$  строк.  $i$ -я из этих строк должна содержать информацию, сможет ли Байтазар в  $i$ -й день присутствовать на двух заседаниях. В случае, если это невозможно, выведите слово "NIE". Иначе выведите слов "ТАК" и номер двух заседаний, на которых может присутствовать Байтазар в соответствующий день. Заседания пронумерованы в том порядке, в котором они описаны во входном файле, нумерация заседаний начинается с единицы. Второе заседание должно начаться как минимум на миллисекунду позже того, как закончилось первое.

Если существует несколько решений, выведите любое из них.

### Examples

standard input	standard output
6 3	ТАК 1 6
3 5 1	NIE
2 4 2	NIE
1 8 1	
6 7 3	
3 5 2	
7 12 1	

## Problem B. Бензин

Input file: *standard input*  
Output file: *standard output*  
Time limit: 1.8 seconds  
Memory limit: 256 mebibytes

Байтазар работает в отделе транспортировки байтландской нефтяной компании Байтойл. В его задачу входит планирование поставок бензина на автозаправочные станции.

Всего в Байтландии  $m$  автомагистралей с двусторонним движением и  $n$  перекрёстков (обозначенных номерами от 1 до  $n$ ). Каждая автомагистраль соединяет некоторые два перекрёстка. Никакая автомагистраль не имеет общих точек с внутренними точками другой автомагистрали. На некоторых перекрёстках размещены заправочные станции, принадлежащие Байтойл.

В распоряжении компании Байтойл имеются бензовозы различной вместимости. Каждый бензовоз вне зависимости от вместимости потребляет 1 литр бензина на километр пути. Водители не могут заправляться бензином из цистерны, которую везут, но могут бесплатно заправляться на любой заправочной станции Байтойл.

Байтазару требуется отвечать на следующие запросы: может ли бензовоз с бензобаком вместимости  $b$  литров доехать от заправочной станции, расположенной на перекрёстке  $x$ , до заправочной станции, расположенной на перекрёстке  $y$ ? Напомним, что бензовоз с бензобаком ёмкостью  $b$  не может проехать более  $b$  километров без заправки.

Напишите программу, которая бы автоматизировала ответы на подобные запросы.

### Input

Первая строка входного файла содержит три целых числа  $n$ ,  $s$  и  $m$  ( $2 \leq s \leq n \leq 200\,000$ ,  $1 \leq m \leq 200\,000$ ) — количество перекрёстков, количество заправочных станций Байтойл и количество дорог в Байтландии. Вторая строка содержит  $s$  попарно различных целых чисел  $c_1, c_2, \dots, c_s$  ( $1 \leq c_i \leq n$ ) — номера перекрёстков, на которых расположены заправки Байтойл.

Следующие  $m$  строк задают дороги в Байтландии.  $i$ -я из этих строк содержит три целых числа  $u_i$ ,  $v_i$  и  $d_i$  ( $1 \leq u_i, v_i \leq n$ ,  $u_i \neq v_i$ ,  $1 \leq d_i \leq 10\,000$ ), обозначающие, что  $i$ -я дорога имеет длину  $d_i$  километров и соединяет перекрёстки  $u_i$  и  $v_i$ . Каждая пара перекрёстков соединена не более, чем одной дорогой.

Следующая строка содержит одно целое число  $q$  ( $1 \leq q \leq 200\,000$ ) — количество запросов. Последующие  $q$  строк задают запросы:  $i$ -я из этих строк содержит три целых числа  $x_i$ ,  $y_i$  и  $b_i$  ( $1 \leq x_i, y_i \leq n$ ,  $x_i \neq y_i$ ,  $1 \leq b_i \leq 2 \cdot 10^9$ ) — исходную точку маршрута, конечную точку маршрута и ёмкость бака бензовоза. Гарантируется, что на обоих перекрёстках  $x_i$  и  $y_i$  размещены заправочные станции Байтойл.

### Output

Выведите  $q$  строк,  $i$ -я из которых должна содержать ответ на  $i$ -й запрос: “ТАК”, в случае, если бензовоз с баком вместимостью  $b_i$  сможет доехать от перекрёстка  $x_i$  до перекрёстка  $y_i$ , и “НІЕ” в противном случае.

## Examples

standard input	standard output
6 4 5	TAK
1 5 2 6	TAK
1 3 1	TAK
2 3 2	NIE
3 4 3	
4 5 5	
6 4 5	
4	
1 2 4	
2 6 9	
1 5 9	
6 5 8	

## Problem C. Цены

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2.5 seconds  
Memory limit: 256 mebibytes

Байтазар работает экспедитором в одном из байтландских ресторанов. Каждый вечер он получает от руководства список продуктов, которые требуется закупить. Продукты должны быть куплены утром следующего дня. Байтазар должен купить ровно одну упаковку каждого из продуктов в списке. При этом суммарные затраты на закупки должны быть минимальными.

Вечером Байтазар узнаёт цены на все требуемые продукты в ближайших оптовых магазинах. Также Байтазару известны затраты на поездку от ресторана до каждого магазина и обратно. Байтазару требуется составить план покупок на утро.

При закупках продуктов Байтазар действует по следующей схеме: он едет от ресторана до магазина, закупает там все продукты, которые планировал, после чего отвозит продукты в ресторан. При этом, в соответствии с санитарными нормами, Байтазар во время одной поездки может покупать продукты только в одном магазине.

Напишите программу, которая определяет минимальные суммарные затраты на закупки продуктов из списка.

### Input

Первая строка входного файла содержит два целых числа  $n$  и  $m$  ( $1 \leq n \leq 100$ ,  $1 \leq m \leq 16$ ) — количество оптовых магазинов и количество наименований продуктов, которые должен купить Байтазар.

Последующие  $n$  строк содержат описания цен в каждом из оптовых магазинов.

Первое число в  $i$ -й из этих строк  $d_i$  ( $1 \leq d_i \leq 1\,000\,000$ ) задаёт стоимость поездки от ресторана до  $i$ -го оптового магазина и обратно. Далее следуют  $m$  целых чисел  $c_{i,1}, c_{i,2}, \dots, c_{i,m}$  ( $1 \leq c_{i,j} \leq 1\,000\,000$ ), где  $c_{i,j}$  задаёт цену на упаковку  $j$ -го продукта в  $i$ -м магазине.

### Output

Выведите одно целое число — минимальные суммарные затраты на закупки всех продуктов из списка.

### Examples

standard input	standard output
3 4 5 7 3 7 9 2 1 20 3 2 8 1 20 1 1	16

### Note

В примере к задаче Байтазар покупает второй продукт в первом магазине, а все остальные — во втором. Таким образом, он избегает необходимости ехать в третий магазин.

## Problem D. Делители

Input file: *standard input*  
Output file: *standard output*  
Time limit: 3.5 seconds  
Memory limit: 256 mebibytes

Вам задана последовательность из  $n$  целых чисел  $a_1, a_2, \dots, a_n$ . Определите количество упорядоченных пар  $(i, j)$ , таких, что  $i, j \in \{1, \dots, n\}$ ,  $i \neq j$  и  $a_i$  является делителем  $a_j$ .

### Input

Первая строка входа содержит одно целое число  $n$  ( $1 \leq n \leq 2\,000\,000$ ). Вторая строка содержит последовательность  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 2\,000\,000$ ).

### Output

Выведите одно целое число — требуемое количество пар.

### Examples

standard input	standard output
5 2 4 5 2 6	6

### Note

В примере к задаче существует 6 пар чисел с требуемыми свойствами:  $(1, 2)$ ,  $(1, 4)$ ,  $(1, 5)$ ,  $(4, 1)$ ,  $(4, 2)$ ,  $(4, 5)$ .

## Problem E. Евклидов Ним

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

Два жителя Байтландии, прозванные Евклидом и Пифагором за их пристрастие к математическим играм, играют в следующую игру. На столе есть кучка из  $n$  камней. Игроки делают ходы по очереди. Если куча содержит не менее  $p$  камней, Евклид может забрать из кучи любое число камней, кратное  $p$ . Если куча содержит ненулевое число камней, меньшее  $p$ , Евклид добавляет ровно  $p$  камней. Ходы Пифагора аналогичны, за исключением того, что вместо  $p$  используется  $q$ .

Победителем считается тот, кто взял последний камень. Первый ход делает Евклид. По заданным  $n$ ,  $p$  и  $q$  выясните, кто победит при оптимальной игре обоих игроков.

### Input

Первая строка входного файла содержит целое число  $t$  ( $1 \leq t \leq 1000$ ) — количество тестовых примеров. Далее заданы сами тестовые примеры, по одному на строку. Каждый тестовый пример состоит из трёх целых чисел  $p$ ,  $q$  и  $n$  ( $1 \leq p, q, n \leq 10^9$ ).

### Output

Для каждого тестового примера в отдельной строке выведите букву 'E', если при правильной игре выигрывает Евклид, 'P', если выигрывает Пифагор, и 'R', если игра будет идти бесконечно.

### Examples

standard input	standard output
4	P
3 2 1	P
2 3 1	E
3 4 5	R
2 4 3	

### Note

В этой игре в первом тестовом примере Евклид должен добавить три камня к куче, после чего Пифагор забирает  $2 \cdot 2 = 4$  камня и выигрывает.

## Problem G. Глобальное потепление

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

Профессор Байтазар готовит отчёт для Всебайтладнской комиссии по изменениям климата. Этот отчёт должен убедить членов комиссии во влиянии деятельности населения на изменения климата в регионе. У профессора имеется много экспериментальной информации, однако для того, чтобы отчёт произвёл соответствующее влияние на комиссию и на прессу, данные должны быть соответственным образом выбраны.

Основная таблица отчёта должна содержать информацию о среднегодичных температурах воздуха. У профессора есть данные по  $n$  последним годам. Он хочет подписать таблицу так: “в году  $r_{min}$  температура была максимальной, а в году  $r_{max}$  она была максимальной, таким образом, очевидно, что . . .”. Однако профессор опасается, что та же самая минимальная или максимальная температура могла быть достигнута несколько раз в соответствующий период, и выводы уже не будут столь очевидными.

Профессор решил, что внесёт в таблицу только часть данных. Его идея заключается в том, что можно выбрать диапазон лет таким образом, чтобы он содержал ровно один год с минимальной температурой в этом диапазоне и ровно один год с максимальной температурой в этом диапазоне. При этом диапазон может не содержать абсолютного максимума за все  $n$  лет, равно как и абсолютного минимума.

Требуется найти такой диапазон, а в случае, если их несколько — самый ранний год, в который может начаться этот диапазон.

### Input

Первая строка входного файла содержит одно целое число  $n$  ( $1 \leq n \leq 500\,000$ ) — количество лет, в течение которых велись наблюдения средней температуры. Следующая строка содержит последовательность из  $n$  целых чисел  $t_1, t_2, \dots, t_n$  ( $-10^9 \leq t_i \leq 10^9$ ). Число  $t_i$  обозначает среднюю температуру за  $i$ -й год.

### Output

Выведите два целых числа  $l$  и  $k$  — максимальную длину диапазона лет, соответствующего требованиям профессора и самый ранний год, когда этот диапазон может начаться.

### Example

standard input	standard output
10 8 3 2 5 2 3 4 6 3 6	6 4

### Note

В таблице будут представлены температуры 5, 2, 3, 4, 6, 3. Этот интервал содержит ровно один год с минимальной средней температурой (2) и ровно один год с максимальной средней температурой (6).

## Problem J. Катакомбы

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2.6 seconds  
Memory limit: 256 mebibytes

Группа спелеологов планирует исследовать недавно обнаруженные катакомбы. Катакомбы состоят из нескольких пещер, занумерованных последовательными целыми числами от 1 до  $n$ . Пещеры соединены  $n - 1$  коридорами таким образом, что из любой пещеры можно пройти в любую другую. Каждый коридор соединяет ровно две пещеры.

В исследовании принимают участие  $m$  спелеологов. Для простоты мы занумеруем их целыми числами от 1 до  $m$ . Каждый спелеолог выдвигает требования к маршруту, по которому он хочет двигаться.  $i$ -й спелеолог собирается начать обследование в пещере  $a_i$ , завершить его в пещере  $b_i$  и пройти не более  $d_i$  коридоров по пути (двукратное прохождение одного коридора считается двумя различными прохождениями).

Байтазар, руководитель экспедиции, требует, чтобы все исследователи могли собраться в определённой точке для того, чтобы обсудить текущее положение дел. Для этого он хочет узнать, сможет ли он выбрать одну из пещер в катакомбах и распланировать маршруты всех спелеологов таким образом, что все они пройдут через эту пещеру (при этом все требования спелеологов к выбранным ими маршрутам должны быть выполнены).

### Input

Первая строка ввода содержит одно целое число  $t$  ( $1 \leq t \leq 1000$ ) — количество тестовых примеров. Далее задаются тестовые примеры. Каждый тестовый пример начинается строкой, содержащей два целых числа  $n$  и  $m$  ( $2 \leq n, m \leq 300\,000$ ) — количество пещер и количество спелеологов, соответственно. Последующие  $n - 1$  строк описывают коридоры. Каждая из этих строк содержит по два целых числа  $u_i, w_i$  ( $1 \leq u_i, w_i \leq n$ ), обозначающие, что пещеры  $u_i$  и  $w_i$  соединены коридором.

В последующих  $m$  строках заданы требования спелеологов к маршрутам.  $i$ -я из этих строк содержит три целых числа  $a_i, b_i, d_i$  ( $1 \leq a_i, b_i \leq n, 1 \leq d_i \leq 600\,000$ ), обозначающие, что спелеолог  $i$  начинает свой маршрут в пещере  $a_i$ , заканчивает в пещере  $b_i$  и хочет пройти не более  $d_i$  коридоров.

Гарантируется, что требования спелеологов осмысленны, то есть что пройти от пещеры  $a_i$  к пещере  $b_i$ , пройдя не более, чем по  $d_i$  коридорам, возможно. Сумма всех  $n$  для всех тестовых примеров не превосходит 300 000. Сумма всех  $m$  для всех тестовых примеров не превосходит 300 000.

### Output

Выведите ровно  $t$  строк.  $i$ -я из этих строк должна содержать ответ к  $i$ -му по порядку тестовому примеру. В случае, если проложить маршруты спелеологов так, чтобы они прошли через одну пещеру, выведите “ТАК”, после чего выведите одно число — номер пещеры, через которую они все пройдут. Иначе выведите “НIE”. Если существует несколько решений, выведите любое.



## Examples

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

## Problem K. Капитан

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2.5 seconds  
Memory limit: 256 mebibytes

Капитан Врунгель и старший помощник Лом возглавляют экспедицию по Байтландскому морю. Всего в Байтландском море  $n$  островов, занумерованных последовательными целыми числами от 1 до  $n$ . Первоначально корабль находится на острове с номером 1. Цель экспедиции — достичь острова с номером  $n$ .

Во время плавания, корабль всё время движется в направлении одной из сторон света: на север, на юг, на запад и на восток. В любой момент на мостике находится или капитан, или первый помощник. Каждый раз, когда корабль поворачивается на  $90^\circ$ , они меняются.

По пути корабль может останавливаться на других островах. После каждой остановки капитан принимает решение, чья вахта (его или помощника) будет первой после остановки.

Иначе говоря, для каждого участка маршрута между двумя островами, один из мореплавателей стоит на мостике, когда корабль идёт на север или на юг, а другой — когда он идёт на запад или на восток. В частности, если маршрут проходит в одном направлении, то смены вахты не происходит.

Врунгель, занятый написанием очередной главы своих мемуаров, собирается распланировать маршрут так, чтобы самому проводить как можно меньше времени на мостике. Предполагается, что корабль движется с постоянной скоростью, равной 1 единице пути в час. Минимизировать длину маршрута не требуется.

### Input

В первой строке входного файла содержится одно целое число  $n$  ( $2 \leq n \leq 200\,000$ ) — количество островов. Введём систему координат с осями, параллельными направлениям на стороны света, каждый остров представим одной точкой. с координатами  $x_i, y_i$  ( $0 \leq x_i, y_i \leq 1\,000\,000\,000$ ); гарантируется, что все острова попарно различны.

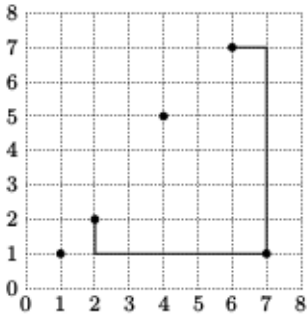
### Output

Выведите одно целое число — минимальное время, которое капитан проведёт на мостике за время маршрута от 1-го острова к  $n$ -му.

### Examples

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

## Note



В примере к задаче Врунгель может выбрать следующий маршрут, указанный на иллюстрации: Во время путешествия от острова 1 (координаты  $(2, 2)$ ) к острову 4 (координаты:  $(7, 1)$ ) капитан стоит на мостике только в течение одного часа, когда корабль плывёт на юг. Во время второго участка пути капитан стоит на мостике только во время движения на восток.

## Problem L. Astroturfing

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

Известная интернет-компания «Дальстолб» разработала онлайн-биржу, на которой частные лица и компании могут продавать некоторые товары. После покупки и тестирования товара покупатель может выставить продавцу рейтинг — целое число от 1 до 10. Для каждого продавца веб-сайт показывает количество голосов  $N$  и среднее значение  $A$ .

Если рейтинг не устраивает компанию, она может или работать лучше, или создать несколько поддельных аккаунтов и поставить с них высокие оценки. Такие действия называются *astroturfing*.

«Дальстолб» предупреждён о такой возможности и для того, чтобы усложнить жизнь нечестным компаниям, ввёл следующие правила: каждый покупатель может рейтинговать одного продавца только один раз; все рейтинги, которые вдвое превышают текущее среднее значение, помечаются для особого исследования.

Ваша задача — вычислить минимальное количество поддельных аккаунтов для того, чтобы поднять рейтинг компании с  $A$  до  $B$ , не запуская особое исследование.

Вычисления на серверах компании ведутся абсолютно точно (то есть без потерь точности).

Для первого примера способ подъёма рейтинга приведён ниже.

Fake account no.	Fake rating	Average rating
1	4	2.40000
2	4	2.66667
3	5	3.00000
4	6	3.37500
5	6	3.66667
6	7	4.00000

### Input

Входной файл содержит целые числа  $N$ ,  $A$ ,  $B$  ( $1 \leq N \leq 10^8$ ,  $1 \leq A < B \leq 9$ ).

### Output

Выведите одно целое число — наименьшее количество поддельных аккаунтов, требуемое для подъёма рейтинга от  $A$  до  $B$ .

### Examples

standard input	standard output
4 2 4	6
182736 1 9	8040384
2 8 9	2

## Problem M. Error Detection

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

Специалисты по Computer Science из ДВГУ используют метод дублирования для обнаружения ошибок при передаче данных по сети. Каждый бит данных дублируется, дубликат следует за оригинальным битом. Каждый байт таким образом трансформируется в сообщение из двух байт. Первый байт сообщения задаёт старшие биты изначального байта, второй — младшие.

Например, число 67 закодировано следующим образом:

1.  $67_{10} = 01000011_2$ ;
2. 01000011 закодировано как 0011000000001111;
3. 0011000000001111 разбито на 00110000 и 00001111;
4.  $00110000_2 = 48_{10}$ ,  $00001111_2 = 15_{10}$ , так что результат равен 4815

Получены два байта. Требуется их расшифровать.

### Input

Входной файл содержит два целых числа в интервале от 0 до 255 — полученные байты.

### Output

Выведите одно целое число — исходный байт, или  $-1$ , если во время передачи произошла ошибка.

### Examples

standard input	standard output
48 15	67
48 7	-1

## Problem N. Хактрис

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

Владивосток Хакатон - мероприятие, где несколько студенческих команд изобретают и создают прототип актуального ПО за несколько дней.

Молодой программист Вася и его команда предложила оригинальную и увлекательную игру "Тетрис". Реализация игры оказалась сложнее, чем они предполагали, поэтому правила игры решили упростить.

Игра происходит на клетчатом прямоугольном поле, состоящим из  $H$  строк и  $W$  столбцов. Каждая клетка или занята или свободна.

Одна частица занимает одну клетку. Новые частицы появляются в некоторой клетке верхней строки и начинают падать вниз. За каждый ход игрок для частицы может выбрать любую свободную клетку из 3 клеток, находящихся строкой ниже, и, соответственно, переместить частицу на 1 клетку влево или на 1 клетку вправо.

Частица перестает падать, когда она достигает нижней строки или когда нижняя клетка занята и игрок решил не двигать частицу левее или правее. После того, как частица останавливается, на доске незамедлительно появляется следующая частица. Больше в игре ничего не происходит — строки не исчезают, очки не подсчитываются и т.п.

Когда Васина команда представляла жюри свою игру, она сломалась и не запустилась снова. Единственное доказательство, что игра работала - лог файл предыдущего запуска игры. Лог файл игры содержит информацию о  $N$  частицах — столбец, где она появилась, и столбец, где она остановилась.

Ваша программа должна по заданному описанию поля и логу игры определить мог ли лог файл быть результатом корректной игры. Если же лог файл не может описывать результат корректной игры, то программа должна вывести первую частицу, которая не могла падать так, как описано в лог файле.

Во втором примере 2, после того, как 6 частиц упали и остановились, и 7-ая появилась, поле выглядит так:

```
.#. . . .  
..#. . . .  
..#. . . .  
..#. . . .  
.###. . #
```

Невозможно переместить частицу в 6-ой столбец, поэтому лог некорректен.

### Input

Входной файл содержит числа  $WHN$ . После этого следуют  $N$  пар чисел  $x_1x_2$  — начальная и конечная колонка каждой частицы ( $1 \leq W, H \leq 10^4, 1 \leq N \leq 10^6, 1 \leq x_1, x_2 \leq W$ ).

### Output

Выходной файл должен содержать одно число — номе первой некорректно поставленной частицы или 0, если описания всех частиц корректны.

## Examples

standard input	standard output
1 10 1 1 1	0
6 5 7 1 3 5 2 6 3 3 3 2 6 4 3 2 6	7