

Problem C. Канатная дорога

Input file: *standard input*
Output file: *standard output*
Time limit: 1 секунда
Memory limit: 256 мебибайт

В городе χ есть своеобразная достопримечательность — старая канатная дорога, которой раньше пользовались рабочие для того, чтобы добраться на работу. Сейчас ее запускают специально для желающих покататься. А не так давно на посадке могла стоять длинная очередь из рабочих разных специальностей. Каждые 5 секунд новый рабочий садился в одиночную кабинку и начинал свой 15-минутный путь на большой высоте.

Дима наткнулся на старую фотографию, на которой запечатлена очередь из N рабочих. Дима смог определить, к какой бригаде относится каждый, по спецодежде. Он замечтался, глядя на выцветший снимок. . . Он представил себе, как рабочие на другом конце канатной дороги ждут своих товарищей и идут вместе трудиться, радостные и полные надежд на счастливое будущее. При этом в воображении Димы каждая бригада трудяг отправлялась в свой цех только тогда, когда прибудет самый последний из них.

Но невозможно было не заметить, что в то время, когда рабочие ждали своих товарищей, станки на заводе простаивали. Дима задумался: а что бы получилось, если бы каждый рабочий пропускал рабочих из других цехов при условии, что это не ухудшало бы время отправления на завод его бригады? Тогда другая бригада могла бы отправиться на завод раньше. Сколько при этом можно было сэкономить времени (на сколько секунд меньше простаивали бы станки во всех цехах суммарно)?

Вам предоставлено описание очереди рабочих на фотографии, определяющее порядок, в котором они стояли, и бригады, к которым они относились. Определите, сколько времени в секундах можно было бы сэкономить, если бы каждый рабочий пропускал рабочих из других бригад вперед в случае, когда это не увеличивало время прибытия на завод его бригады.

Input

В первой строке входа содержится число N — число рабочих в очереди. $1 \leq N \leq 25000$. Вторая строка содержит строку из N символов латинского алфавита или цифр. i -й символ в строке определяет бригаду i -го рабочего в очереди. Рабочие, которым соответствуют равные символы, относятся к одной бригаде.

Output

На выходе число секунд, которые можно было сэкономить.

Examples

standard input	standard output
6 AABABB	15

Problem D. Рассадка

Input file: *standard input*
Output file: *standard output*
Time limit: 2 секунды
Memory limit: 256 мегабайт

В одном из детских садов города χ популярны игры, во время которых дети сидят по кругу так, что у каждого ребенка есть ровно два соседа (слева и справа). Каждая такая игра начинается с долгого и непростого процесса рассадки. Дело в том, что некоторым детям важно, кто является их соседями (они хотят сидеть рядом со своими друзьями), для других же этот вопрос не принципиален.

Помогите воспитателю одной из групп детского сада. Всего в группе N детей, и пусть для простоты каждый ребенок имеет номер от 1 до N . Дети поделились с воспитателем своими предпочтениями. Воспитателю известно о W фактах вида «Ребенок A_i сказал, что хочет сидеть рядом с ребенком B_i » (однако это не значит, что непременно B_i тоже высказывал желание занять место рядом с A_i). Каждый ребенок заявлял не более двух пожеланий. Воспитатель хочет узнать, возможно ли вообще так рассадить детей, чтобы пожелания всех были исполнены.

Input

Тестовые данные включают набор из не более 100 отдельных тестов.

Каждый тест начинается со строки, содержащей два разделенных пробелом числа N ($3 \leq N \leq 10^9$) и W ($0 \leq W \leq 10^5$). Далее следуют W строк, описывающих пожелания ребят. В каждой строке записаны через пробел числа A_i и B_i ($1 \leq A_i, B_i \leq N$, $A_i \neq B_i$), которые соответствуют одному пожеланию. Каждое число встречается в качестве A_i не более чем дважды.

За последним тестом следует строка, содержащая два нуля. Ее обрабатывать не нужно.

Гарантируется, что сумма всех W для одного набора тестов не превосходит 10^5 .

Output

Для каждого теста выведите в отдельной строке «YES» (без кавычек), если рассадить детей в соответствии с их желаниями можно, и «NO» в противном случае.

Examples

standard input	standard output
3 5 2 1 2 3 1 2 3 1 1 3 123456789 0 4 3 1 2 2 3 3 1 0 0	YES YES NO

Problem E. Метро

Input file: *standard input*
Output file: *standard output*
Time limit: 3 секунды
Memory limit: 256 мегабайт

Метро города Нью-Прудный состоит из n станций. Расположение каждой станции можно задать координатами в декартовой системе координат: x_i, y_i, z_i . При создании подземки для помощи привлекли учёных с дружественной нам звезды. Инопланетяне использовали неизвестные технологии и создали очень сложную систему дорог. Чтобы добраться от станции i до станции j , требуется $\rho(i, j) = |x_i - y_j| + |y_i - z_j| + |z_i - x_j|$ миллисекунд. Учёных со всей Земли особенно удивляет, что $\rho(i, i)$ может быть больше нуля и $\rho(i, j)$ может быть не равно $\rho(j, i)$.

Нью-Прудненцы пользуются метро уже несколько лет, но так и не поняли, сколько времени добираться между станциями. Известная рок-группа «Високосная секунда» решила написать новую песню про две самые удалённые станции метро. Помогите группе порадовать слушателей новой песней, найдите номера двух самых удалённых станций.

Стоит заметить, что в метро нельзя делать пересадки, и группу интересуют две различные станции.

Input

В первой строке задано натуральное число n — количество станций в Нью-Прудненском метро ($2 \leq n \leq 1\,000\,000$). Далее на n различных строках записаны по три числа — координаты каждой станции x_i, y_i, z_i . Координаты по модулю не превосходят 10^5 .

Output

В первой строке выведите время в миллисекундах между двумя наиболее удалёнными станциями, то есть максимальное $\rho(x, y)$ ($x \neq y$). Во второй строке выведите два различных числа x и y — номера двух наиболее удалённых станций. Станции нумеруются с единицы в порядке задания их во входном файле. Если ответов несколько — выведите любой.

Examples

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

Problem F. Пекарня

Input file: *standard input*
Output file: *standard output*
Time limit: 1 секунда
Memory limit: 256 мебибайт

Пекарня города Нью-Прудного способна произвести до c_i буханок хлеба в i -ый день. При этом каждая буханка хлеба обойдется пекарне в f_i условных Нью-Прудненских долларов. Известно, что для обеспечения города продовольствием в день требуется d_i буханок хлеба, которые успешно будут употреблены населением. Оставшиеся буханки отправляются на склад и могут быть использованы в следующие дни. На складе можно в ночь с i ого на $i + 1$ день хранить не более g_i буханок, при этом хранение каждой буханки хлеба на складе обойдется пекарне в e_i Нью-Прудненских долларов.

Ваша задача — определить наименьшее количество денег, которое пекарня потратит на производство и хранение, чтобы обеспечить Нью-Прудный хлебом в течение n дней.

Input

В первой строке содержится натуральное число n ($1 \leq n \leq 10^5$). Далее в n строках содержатся числа c_i, f_i, d_i . Затем в $n - 1$ строках содержатся числа g_i, e_i ($0 \leq c_i, f_i, d_i, g_i, e_i \leq 10^9$).

Output

Выведите единственное число — минимальное количество денег, необходимое пекарне. Если удовлетворить требованиям города невозможно, выведите -1.

Examples

standard input	standard output
3 10 1 1 2 2 2 10 10 8 7 2 6 5	73

Problem G. Палиндромы Фибоначчи

Input file: *standard input*
Output file: *standard output*
Time limit: 1 секунда
Memory limit: 256 мегабайт

Рассмотрим следующие строки, состоящие из цифр 0 и 1:

$S_0 = "0"$
 $S_1 = "01"$
 $S_2 = S_1S_0 = "010"$
 $S_3 = S_2S_1 = "01001"$
 $S_4 = S_3S_2 = "01001010"$
 $S_5 = S_4S_3 = "0100101001001"$
...
 $S_n = S_{n-1}S_{n-2}$
...

Как видно из определения, строка S_{n-1} является префиксом строки S_n для любого $n > 0$. Значит, существует бесконечная строка S_∞ , являющаяся *пределом* строк S_n , то есть такая, что для любого $n \geq 0$ строка S_n является префиксом S_∞ .

(Если предыдущее определение не удалось понять, то можете считать, что S_∞ — это просто S_n для достаточно большого n , скажем, $n = 10^{100}$)

Строка S_∞ называется *строкой Фибоначчи*.

Вам даны несколько подстрок строки Фибоначчи. Выясните, какие из них являются палиндромами. Строка называется *палиндромом*, если её первый символ совпадает с последним, второй — с предпоследним, и так далее.

Input

В первой строке находится натуральное число n — количество подстрок ($1 \leq n \leq 10^4$). Далее заданы сами строки. Каждая из них задана двумя числами *start* и *len* и соответствует подстроке строки S_∞ , начинающейся с позиции *start* (позиции нумеруются с единицы) и состоящей из *len* символов ($1 \leq start, len \leq 10^9$).

Output

Выведите n строк, по одной для каждой входной подстроки. Для очередной подстроки выведите "ТАК", если она является палиндромом, и "НЕТ" в противном случае.

Example

standard input	standard output
4	ТАК
2 4	НЕТ
3 5	ТАК
3 7	ТАК
1 11	

Problem H. Художник

Input file: *standard input*
Output file: *standard output*
Time limit: 3 секунды
Memory limit: 256 мебибайт

Художник Вася Ниимович рисует абстрактные картины. Новая его картина выглядит как прямая, части которой раскрашены в разные цвета. Рисовал он её так. Вначале он нарисовал отрезок первого цвета от точки l_1 до точки r_1 . Потом отрезок второго цвета от l_2 до r_2 , и так далее. Последний отрезок, нарисованный им, проходил от l_n до r_n и имел цвет с номером n .

Все отрезки были разных цветов, однако некоторые оказались полностью закрашены другими. Тогда Вася задумался: возможно, если бы он рисовал отрезки в тех же местах, но в другом порядке, то различных цветов на картине было бы больше? Помогите ему выбрать такой порядок рисования отрезков, при котором после того, как он нарисует все отрезки, количество видимых цветов на картине будет максимально. Отрезки считаются замкнутыми.

Input

В первой строке написано целое число n ($1 \leq n \leq 300$). В следующих n строках будут написаны по два целых числа через пробел. В i -ой из этих строк находятся числа l_i и r_i ($-1\,000\,000\,000 \leq l_i < r_i \leq 1\,000\,000\,000$).

Output

В первой строке выведите количество цветов, которые будут видны при оптимальном порядке рисования. Во второй строке должно быть написано n чисел — i -ое число обозначает, какой отрезок надо нарисовать i -ым для достижения оптимального результата. Отрезки нумеруются, начиная с единицы, в том же порядке, в котором они заданы во входном файле.

Example

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