Problem A. Разрезание графа

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

Дан неориентированный граф. Над ним в заданном порядке производят операции следующих двух типов:

- `cut` — разрезать граф, то есть удалить из него ребро;
- `ask` — проверить, лежат ли две вершины графа в одной компоненте связности.

Известно, что после выполнения всех операций типа `cut` рёбер в графе не осталось. Найдите результат выполнения каждой из операций типа `ask`.

Input

Первая строка входного файла содержит три целых числа, разделённые пробелами — количество вершин графа n, количество рёбер m и количество операций k ($1 \le n \le 50\,000$, $0 \le m \le 100\,000$, $m \le k \le 150\,000$).

Следующие m строк задают рёбра графа; i-ая из этих строк содержит два числа u_i и v_i ($1 \le u_i, v_i \le n$), разделённые пробелами — номера концов i-го ребра. Вершины нумеруются с единицы; граф не содержит петель и кратных рёбер.

Далее следуют k строк, описывающих операции. Операция типа `cut` задаётся строкой "`cut` u v" ($1 \le u, v \le n$), которая означает, что из графа удаляют ребро между вершинами u и v. Операция типа `ask` задаётся строкой "`ask` u v" ($1 \le u, v \le n$), которая означает, что необходимо узнать, лежат ли в данный момент вершины u и v в одной компоненте связности. Гарантируется, что каждое ребро графа встретится в операциях типа `cut` ровно один раз.

Output

Для каждой операции `ask` во входном файле выведите на отдельной строке слово "`YES`", если две указанные вершины лежат в одной компоненте связности, и "`NO`" в противном случае. Порядок ответов должен соответствовать порядку операций `ask` во входном файле.

Example

standard input	standard output
3 3 7 1 2 2 3 3 1 ask 3 3 cut 1 2 ask 1 2 cut 1 3 ask 2 1 cut 2 3 ask 3 1	YES YES NO NO

Problem B. Минимальный каркас

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

Требуется найти в связном графе остовное дерево минимального веса.

Input

Первая строка входного файла содержит два натуральных числа n и m – количество вершин и ребер графа соответственно ($1 \le n \le 100000$, $0 \le m \le 100000$). Следующие m строк содержат описание ребер по одному на строке. Ребро номер i описывается тремя натуральными числами b_i, e_i и w_i — номерами концов ребра и его вес соответственно ($1 \le b_i, e_i \le n$, $0 \le w_i \le 100000$).

Гарантируется, что граф связный.

Output

Выведите единственное число – вес минимального остовного дерева.

Examples

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

Problem C. День Объединения

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

В Байтландии есть целых n городов, но нет ни одной дороги. Король страны, Вальдемар де Беар, решил исправить эту ситуацию и соединить некоторые города дорогами так, чтобы по этим дорогам можно было добраться от любого города до любого другого. Когда строительство будет завершено, король планирует отпраздновать День Объединения. К сожалению, казна Байтландии почти пуста, поэтому король требует сэкономить деньги, минимизировав суммарную длину всех построенных дорог.

Input

Первая строка входного файла содержит натуральное число n ($1 \le n \le 5\,000$) — количество городов в Байтландии. Каждая из следующих n строк содержит по два целых числа x_i, y_i — координаты i-го города ($-10\,000 \le x_i, y_i \le 10\,000$). Никакие два города не расположены в одной точке.

Output

Первая строка выходного файла должна содержать минимальную суммарную длину дорог. Выведите число с абсолютной точностью не менее 10^{-3}.

Example

standard input	standard output
6 1 1 7 1 2 2 6 2 1 3 7 3	9.6568542495

Problem D. Шпионы

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

Вы — глава агенства, в котором работает N сотрудников — шпионы с кодовыми номерами от 1 до N. Шпионы были посланы в различные страны и получили кусочки важной информации. Ваша задача состоит в следующем:

1. Организовать встречи между некоторыми шпионами. На каждой такой встрече ровно два шпиона оказываются в одном месте и обмениваются всей информацией, которую получили сами или узнали от других шпионов на предыдущих встречах. Каждую из возможных встреч сложно и затратно организовать, поэтому вам известны все стоимости.

2. После того, как все встречи состоялись, выбирается группа шпионов, и они посылаются на задание по спасению мира. Для каждого шпиона известно, сколько денег он изведет, если его включить в эту группу. Задание должно быть выполнено, поэтому очень важно, чтобы выбранные шпионы все вместе обладали всей информацией, полученной агенством.

Найдите минимальную цену, за которую можно спасти мир.

Input

Первая строка содержит натуральное число N — количество шпионов ($2 \le N \le 1000$). Каждая из следующих строк содержит N натуральных чисел, не превосходящих 10^6. Число в k-ой строке и m-ом столбце обозначает стоимость встречи между шпионами k и m и совпадает с числом в m-ой строке и k-ом столбце (если k=m, то соответствующее число будет равно 0). Следующая строка содержит n чисел, i-ое из них равно стоимости послать i-го шпиона на задание. Все стоимости являются положительными целыми числами.

Output

Выведите одно число - минимальную стоимость спасения мира.

Examples

standard input	standard output
3 0 6 9 6 0 4 9 4 0 7 7 7	17
3 0 17 20 17 0 10 20 10 0 15 9 12	34
5 0 3 12 15 11 3 0 14 3 20 12 14 0 11 7 15 3 11 0 15 11 20 7 15 0 5 10 10 10 10	28

Problem E. Широчайшие лодки

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

В Стране Воды есть N озер, пронумерованных от 1 до N, и M каналов между ними. Ширина W_{ij} канала между каждой парой озер (i, j), где есть канал, известна. Перемещение по каналам возможно в обоих направлениях.

Пароходная компания подготовила список из K пар озер, между которыми нужно постоянно держать связь. Гарантируется, что от любого озера можно добраться по каналам до любого другого (возможно, через другие озера).

Для каждой пары озер из списка компании Вам нужно узнать наибольшую ширину лодки, которая сможет добраться от одного озера этой пары до другого.

Input

Первая строка содержит 3 целых числа N, M и K ($1 \le N \le 1000$, $1 \le M \le 10^5$, $1 \le K \le 10^4$, $1 \le w_{ij} \le 200$).

Каждая из следующих M строк содержит по 3 числа i, j, w ($1 \le i, j \le N$). Это значит, что между озерами i и j есть канал ширины w.

Затем идут K строк, каждая из которых содержит 2 числа i и j ($1 \le i, j \le N$).

Output

Для каждой из последних K строк входного файла выведите в отдельной строке, какую наибольшую ширину может иметь лодка, которая сможет проехать между озерами с номерами в этом запросе.

Examples

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

Problem F. Всем чмоки в этом чатике!

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

Сегодня Мэри, как программисту социальной сети «Телеграфчик», предстоит реализовать сложную систему управления чатами.

Задача Мэри усложняется тем, что в социальную сеть «Телеграфчик» внедрена продвинутая система шифрования «ZergRus», простая, как всё гениальное. Суть её в том, что в системе хранится одна переменная zerg, которая принимает значения от 0 (включительно) до $p = 10^6 + 3$ (исключая p) и меняется в зависимости от событий в системе.

В социальной сети всего n пользователей ($1 \le n \le 10^5$). В начале дня каждый пользователь оказывается в своём собственном чате, в котором больше никого нет. Переменная zerg в начале дня устанавливается равной 0.

В течение дня происходят события типов:

1. Участник с номером $(i + zerg) \bmod n$ посылает сообщение всем участникам, сидящим с ним в чате (в том числе и себе самому), при этом переменная zerg заменяется на $(30 \cdot zerg + 239) \bmod p$.
2. Происходит слияние чатов, в которых сидят участники с номерами $(i + zerg) \bmod n$ и $(j + zerg) \bmod n$. Если участники и так сидели в одном чате, то ничего не происходит. Если в разных, то чаты объединяются, а переменной zerg присваивается значение $(13 \cdot zerg + 11) \bmod p$.
3. Участник с номером $(i + zerg) \bmod n$ хочет узнать, сколько сообщений он не прочитал, и прочитать их. Если участник прочитал q новых сообщений, то переменной zerg присваивается значение $(100\,500 \cdot zerg + q) \bmod p$.

Вы поможете Мэри реализовать систему, обрабатывающую эти события?

Input

В первой строке входного файла записаны натуральные числа n ($1 \le n \le 10^5$) "— число пользователей социальной сети. и m ($1 \le m \le 3 \cdot 10^5$) "— число событий, произошедших за день. В следующих m строках содержится описание событий. Первое целое число в строке означает тип события t ($1 \le t \le 3$). Если $t = 1$, далее следует число i ($0 \le i < n$), по которому можно вычислить, какой участник послал сообщение. Если $t = 2$, далее следуют числа i и j ($0 \le i, j < n$), по которым можно вычислить номера участников, чаты с которыми должны объединиться. Если $t = 3$, далее следует число i ($0 \le i < n$), по которому можно вычислить номер участника, желающего узнать, сколько у него сообщений, и прочитать их.

Output

Для каждого события типа 3 нужно вывести число непрочитанных сообщений у участника.

Examples

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