Problem A. Spanning tree

Input file:	`system input`
Output file:	`system output`
Time limit:	2 seconds
Memory limit:	256 megabytes

You are given points on the plane which are the vertices of the complete graph. The weight of an edge equals to the distance between the points. You have to find the spanning tree of minimal weight.

Input

The first line of the input contains one integer n ($1 \le n \le 5000$) — the number of vertices in the graph. The next n lines contain two integers each x_i, y_i ($-10\,000 \le x_i, y_i \le 10\,000$) — the coordinates of the i-th point. None of the two points coincide.

Output

The sole line of the input should contain one real number – the weight of the minimal spanning tree.

Examples

system input	system output
3 0 0 1 0 0 1	2

Problem B. Spanning tree 2

Input file:	`system input`
Output file:	`system output`
Time limit:	2 seconds
Memory limit:	256 megabytes

You have to find the minimal spanning tree in the connected graph.

Input

The first line of the input contains two integers n and m ($1 \le n \le 20\,000, 1 \le m \le 100\,000$) — the number of vertices and edges, correspondingly. The next m lines contains three integers each b_i, e_i and w_i ($1 \le b_i, e_i \le n, 0 \le w_i \le 100\,000$).

Output

The sole line of the output should contain one integer — the weight of the minimal spanning tree.

Examples

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

Problem C. Dense spanning tree

Input file:	`system input`
Output file:	`system output`
Time limit:	2 seconds
Memory limit:	256 megabytes

You have to find the spanning tree in which the difference between the maximal and the minimal edge is minimal.

Input

The first line of the input contains two integers n and m ($1 \le n \le 1\,000, 1 \le m \le 10\,000$) — the number of vertices and edges, correspondingly. The next m lines contains three integers each b_i, e_i and w_i ($1 \le b_i, e_i \le n, 0 \le |w_i| \le 10^9$).

Output

If there exists a spanning tree, the first line should contain `YES` and the second line should contain one integer — the minimal difference between maximal and minimal edges in the spanning tree. Otherwise, the first line should contain `NO`.

Examples

system input	system output
4 5 1 2 1 1 3 2 1 4 1 3 2 2 3 4 2	YES 0

Problem D. Dense tree

Input file:	`system input`
Output file:	`system output`
Time limit:	2 seconds
Memory limit:	256 megabytes

You have to perform two types of queries:

- '1 x y w' — insert an edge $x - y$ with weight w to the graph;
- '2 x' — count the sum of the weights of edges inside the connected component with vertex x.

Input

The first line contains two integers n and m ($1 \le n, m \le 10^6$) — the number of vertices in the graph and the number of queries. Then each of the next m lines contains a query. The query could be of two types: '1 x y w' ($1 \le x, y \le n, 1 \le w \le 10^3$) or '2 x' ($1 \le x \le n$).

Output

For each of the request of the second kind output the answer on a separate line.

Examples

system input	system output
6 10 2 1 1 1 2 1 2 1 1 2 4 2 2 1 1 1 4 3 2 1 1 3 5 3 2 5 2 6	0 1 3 6 3 0

Problem E. Cutting the graph

Input file:	`system input`
Output file:	`system output`
Time limit:	2 seconds
Memory limit:	256 megabytes

You are given an undirected graph. You have to perform two types of requests:

- `cut` u v — remove an edge $u - v$ from the graph;
- `ask` u v — check whether two vertices u and v are connected.

It is known that after all the requests there are no edges left in the graph. Find the results of the requests of the second type.

Input

The first line of the input contains three integers n, m and k ($1 \le n \le 50\,000, 0 \le m \le 100\,000, m \le k \le 150\,000$) — the number of vertices, the number of edges and the number of requests.

Each of the next m lines contains an edge $u - v$ ($1 \le u, v \le n$). The graph does not contain loop and mutiple edges.

Each of the next k lines contains a request: "`cut` u v" ($1 \le u, v \le n$) or "`ask` u v" ($1 \le u, v \le n$).

Output

For each request of the second type you should output on a separate line "`YES`", if the vertices are connected, or "`NO`", otherwise.

Examples

system input	system 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 F. Transport

Input file:	`system input`
Output file:	`system output`
Time limit:	2 seconds
Memory limit:	256 megabytes

You become in charge of transportation in the country. Currently there are no possibility to travel between the cities, and you have to build everything from scratch. For that he could build the roads between the cities and build the airports in them.

To build a road of length L you have to pay $R \times L$. The road could be build only from one city directly to another. The construction of an airport costs A.

You want to make that there is a possibility to get from one city to another. You can get from the city X to the city Y if:

- there exists a direct road from X to Y;
- or X and Y contain airports;
- or there exists a city Z, such that Z is reachable from X and Y is reachable from Z.

Given the coordinates of the cities and constants R and A, find the minimal cost to build the transportation network.

Input

The first line of the input contains one integer n ($1 \le n \le 150$) — the number of cities. The second line contains x-coordinates of the cities. The third line contains y-coordinates of the cities. All the coordinates are from 0 to 10^6. The fourth line contains a real R, and the fifth line contains a real A.

Output

The sole line of the output should contain the answer with the precision 10^{-6}.

system input	system output
4 0 0 400 400 0 100 0 100 1.0 150.0	500.0000000000
5 0 0 400 400 2000 0 100 0 100 2000 1.0 500.0	1600.0000000000
8 0 100 200 300 400 2000 2100 2200 0 100 200 300 400 2000 2100 2200 0.5 200.0	824.2640687119

Problem G. Transport 2

Input file:	`system input`
Output file:	`system output`
Time limit:	2 seconds
Memory limit:	256 megabytes

You want again to build a transportation network. You find which roads could be build and how much they cost. Sometimes, you could build autobahns instead of roads. Each autobahn costs c times more than the normal road.

You want to build a transportation network that it costs no more than k; it is possible to get from one city to another; and it contains as much autobahns as possible.

Input

The first line of the input contains four integers n, m, k and c ($1 \le n, m \le 100\,000$, $1 \le k \le 10^9$, $1 \le c \le 1000$) — the number of cities, the number of possible roads, the total amount of money and the coefficient.

Each of the next m lines contains the description of the road a, b and l ($1 \le a, b \le n$, $1 \le l_i \le 10^6$) — the cities that are connected and the amount of money it costs.

There are no loop roads and no two roads connect the same cities.

Output

If it is impossible to build such transportation network output "Impossible". Otherwise, the first line of the output should contain two integers p and q — the number of roads and autobahns. The second line should contain p integers — the identifiers of the roads in increasing order. The third line should contain q integers — the identifiers of the autobahns in increasing order.

The roads are enumerated in the order of the input file. If there are several solutions, output any of them.

Examples

system input	system output
4 2 10 2 1 2 3 3 4 5	Impossible
4 4 10 2 1 2 3 3 4 5 1 3 1 3 2 1	1 2 2 3 4

Problem H. Cycle

Input file:	`system input`
Output file:	`system output`
Time limit:	2 seconds
Memory limit:	256 megabytes

There are n cities and m roads in the town. The road i contains c_i coins. You choose some cycle in the graph that contains each road at most once. Suppose that the chosen cycle consists of roads with coins $c_1, \ldots, c_k$ on them, then you get $\min(c_1, \ldots, c_k) + \max(c_1, \ldots, c_k)$ coins.

Find the cycle that will give you the maximal number of coins.

Input

The first line of the input contains two integers n and m ($1 \le n, m \le 100\,000$) — the number of cities and the number of roads.

The next m lines contain the description of the roads. i-th line contains three integers v, u, w ($1 \le v, u \le n, v \ne u, 0 \le w \le 10^9$) — the identifiers of cities that are connected by the i-th road and the number of coins on the i-th road. There are could be multiple roads between the pair of cities.

Output

The sole line of the output should contain the answer on the problem. If there is no cycle at all, output 0.

Examples

system input	system output
3 3 1 2 1 2 3 1 3 1 1	2
4 4 1 2 1 2 3 2 3 1 1 1 4 100	3
4 5 1 2 2 2 3 1 3 1 1 1 4 2 4 2 2	4
2 1 1 2 1	0