Problem A - Abnormal 97ers

New students have just started their education at the Amirkabir University of Technology. Because of the financial crisis, they have decided to buy a taxi and work on afternoons. Since they are intelligent, they want to maximize their income. They only work on the Valie-asr street. Suppose the Valie-asr is a straight street and at each point, a car can turn around 180 degrees and change its direction along the street. MeHdi, one of the new students is given a request to take two passengers and drop them off to their destinations. The order of picking up and dropping the passengers off is completely up to MeHdi, i.e. let say he has to pick up passenger A and B and drop them off at their destinations, he can take both of them at first and drop them off at their destination, or he can take A and drop him/her and then take B and drop him/her, or vice versa. He only wants to minimize the total distance he has to drive to drop both passengers at their destinations.

Input Format

The first line of input contains an integer x_m, the starting point of MeHdi. The second line of input contains two space-separated integers x_1, x_2, the starting point and end point of the first passenger. The third line of input contains two space-separated integers x_3, x_4, the starting point and end point of the second passenger.

Constraints

$1 \leq x_m, x_1, x_2, x_3, x_4 \leq 100{,}000$

Output Format

Print one line of output containing the minimum distance MeHdi has to drive to pick up and drop off both of the passengers at their destinations.

Sample Input	Sample Output
1 2 4 3 5	4

Problem B - Algo

Mohsen has taken algorithm class this semester. One exercise all students have to do is presenting an NP-Complete problem during the semester. Their teacher, Dr. Tabesh has made an interesting process of selecting the problem which they suppose to present. Each student should send an email to Dr. Tabesh containing 3 NP-Complete problems. Then Dr. Tabesh will iterate through students choices one by one, according to their sent time, and give them the first problem in their list that has not been given to any other students before. If all of the 3 choices of a student have been given to the other students previously, then he/she is unlucky and has to take the course next semester again!

Input Format

The first line of input contains an integer n. Each of the next n lines contains 3 space-separated NP-Complete problems.

Constraints

$1 \leq n \leq 100$

Any NP-Complete problem name only consists of at most 20 uppercase and lowercase English letters.

Output Format

Print n lines of output, each containing the problem which has been given to the i-th student, or print "Unlucky".

Sample Input	Sample Output
4 BandwidthProblem ClosestString FlowShopScheduling BandwidthProblem GraphColoring FlowShopScheduling GraphColoring BandwidthProblem FlowShopScheduling FlowShopScheduling BandwidthProblem GraphColoring	BandwidthProblem GraphColoring FlowShopScheduling Unlucky

Problem C - Distinct

In computer science, **Prim's algorithm** is a greedy algorithm that finds a minimum spanning tree for a weighted undirected graph. This means it finds a subset of the edges that form a tree that includes every vertex, where the total weight of all the edges in the tree is minimized. The algorithm operates by building this tree one vertex at a time, from an arbitrary starting vertex, at each step adding the cheapest possible connection from the tree to another vertex. Let's assume our graph has n vertices and m edges and vertices are indicated by integers 1 to n.

The algorithm may informally be described as performing the following steps:

1. Initialize a tree with *vertex 1*.
2. Grow the tree by one edge: of the edges that connect the tree to vertices not yet in the tree, find the minimum-weight edge, and transfer it to the tree. In case of a tie, choose the edge that adds the vertex with *minimum index*.
3. Repeat step 2 (until all vertices are in the tree).

Dijkstra's algorithm is an algorithm for finding the shortest paths between nodes in a graph, which may represent, for example, road networks.

Let the node at which we are starting be called *the initial node*, in this problem we choose node *1 to be the initial node*. Let the distance of node *Y* be the distance from the initial node to *Y*. Dijkstra's algorithm will assign some initial distance values and will try to improve them step by step.

1. Mark all nodes unvisited. Create a set of all the unvisited nodes called the *unvisited set*.
2. Assign to every node a tentative distance value: set it to zero for our initial node and to infinity for all other nodes. Set the initial node as current.
3. For the current node, consider all of its unvisited neighbors and calculate their *tentative* distances through the current node. Compare the newly calculated *tentative* distance to the currently assigned value and assign the smaller one. For example, if the current node *A* is marked with a distance of 6, and the edge connecting it with a neighbor *B* has length 2, then the distance to *B* through *A* will be 6 + 2 = 8. If B was previously marked with a distance greater than 8 then change it to 8. Otherwise, keep the current value.
4. When we are done considering all of the unvisited neighbors of the current node, mark the current node as visited and remove it from the *unvisited set*. A visited node will never be checked again.
5. Select the unvisited node that is marked with the smallest tentative distance, set it as the new "current node", and go back to step 3. *In case of tie select the node with the smallest index.*

If you take a closer look at the definition of these two algorithms, both of them generate a tree from the given graph (In Dijkstra's algorithm, set the last node that updated distance of node Y as $parent_y$, if you select all edges $(node, parent_{node})$ they will form a tree). Now we want you to write a program that generates a connected graph with n nodes and m edges that under these two algorithms generates different trees, if it is possible. **Two trees are considered different** if there exist two vertices which are connected by an edge in one and not connected in the other.

Input Format
The only line of input contains two space-separated integers, n and m - the number of nodes and number of edges, respectively.

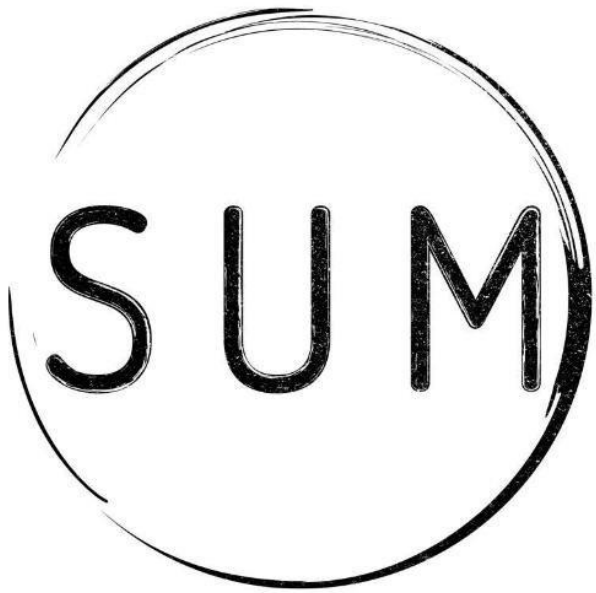
Constraints
$1 \leq n \leq 100$
$0 \leq m \leq n.(n-1)/2$
$1 \leq x, y \leq n \; and \; x \neq y$
$0 < cost < 1{,}000{,}000$

Output Format
If it is possible, print m lines of output each containing three space separated integers, x, y, cost, which means there is an edge between node x and y with length of cost, otherwise print IMPOSSIBLE. Please note that no two edges must have the same end points.

Sample Input	Sample Output
`5 4`	`IMPOSSIBLE`

Problem D - Tuple Sum

We couldn't make any story for this problem! So let's get right to the problem. For a given increasing sequence $A\ (a_1,\ldots,a_n)$ of positive integers where for every i in $[1,n]$, $a_i \le a_{i+1}$, function $f(k,m)$ calculates the number of different $B\ (b_1,\ldots,b_n)$ sequences, such that for every i in $[1,n]$, $0 \le bi \le ai$ and also $(\sum_{i=1}^{n} b_i)\ MOD\ m = k$.



Input Format
The first line of input contains three positive integers n, m and k. Second line of input contains n space-separated integers, $a_1, a_2, \ldots, a_n$.

Constraints
$1 \le n \le 2000$
$0 \le m \le 2000$
$0 \le k < m$
$1 \le ai \le 10000$

Output Format
Print one line of output containing $f(k)$ modulo 1,000,000,007.

Sample Input	Sample Output
2 10 3 5 7	4

Problem E - Taxi Line

Reza lives in a city where there is only one road from the leftmost part of the city to the rightmost part of the city, let's call leftmost point station L and rightmost point station R.

This road consists of exactly **K** segments. There are **M** people who want to get to their offices every morning, all of them live near station L or near station R. They need to get to their offices using one of the only **N** taxis in this city. Every morning some taxis start at station L and the remaining taxis start at station R. Each taxi has a capacity of up to 4 passengers. Taxis are numbered from 1 to **N**, every morning they are lined up according to their corresponding taxi number, i.e. taxis with numbers 1, 4, 7 start at station L and taxis with numbers 2, 3, 5, 6 start at station R. Passengers are numbered from 1 to **M** and they stand in a queue according to their numbers. For each passenger, we know he/she will get off on the intersection of two consecutive segments, excluding L and R (L and R cannot be a destination for any of the passengers). After all passengers of one taxi are gotten off, the taxi drives to the nearest of L and R, **in case of a tie drives to station L**, and parks at the end of the queue in that station. Your task is to write a program to determine which taxi does each passenger use to get to his/her destination? For simplicity, you can assume that at any given time only one taxi is working and all other taxis are waiting. Taxis start to take passengers according to their number, i.e. first taxi number 1, then taxi number 2, and so on. If some taxi has no passenger in its station, then the next taxi according to its number starts taking passengers.

Input Format

The first line of input contains three space-separated integers, K, N and M, the number of segments of a road, the number of taxis and the number of passengers, respectively.
Each of the next K+1 lines contains two integers x and y, corresponding to coordinates of the points of the segments, from left to right.
The next N lines each contains either 0 or 1, the i-th line shows that taxi with number i starts at station L or station R, 0 denotes station L, 1 denotes station R.
Each of the next M lines contains 3 numbers, S, x and y. S is 0 or 1, (passenger starts at L or R) and (x,y) is the coordinate of his/her destination, x and y are integers.

Constraints

$1 \leq k, n, m \leq 1000$

$0 \leq xi, yi \leq 50{,}000$

Output Format

Print m lines of output, each containing a taxi number, the j-th line shows the taxi number which j-th passenger uses to get to his/her destination, or -1 if he/she cannot get to his/her destination.

Sample Input	Sample Output
3 4 18 7324 1543 7403 1870 7733 622 8118 1373 1 1 0 1 1 7733 622 1 7733 622 1 7403 1870 0 7403 1870 0 7733 622 0 7733 622 1 7733 622 1 7733 622 1 7733 622 0 7733 622 1 7733 622 1 7403 1870 0 7733 622 0 7403 1870 1 7403 1870 0 7733 622 0 7403 1870 0 7403 1870	1 1 1 3 3 3 1 2 2 3 2 2 1 1 4 1 1 2

Problem F - Snapp Wifi

Snapp, the transportation network company, is going to add some wifi stations in Tehran. Each taxi in the Snapp network will have access to wifi if it is in a certain area that is covered by a wifi station. A wifi station's shape is like a simple polygon. In geometry a simple polygon is a flat shape consisting of straight, non-intersecting line segments or "sides" that are joined pair-wise to form a closed path. If the sides intersect then the polygon is not simple.
Taxis do several trips every day. Each trip consists of a starting point, some segments and a destination. Starting point and end point are strictly in a wifi station. A taxi moves along each segment one by one until reaching destination. Now Snapp authorities want to check for some trips, the maximum distance the taxi gets from any wifi station along its path. Wifi stations do not touch or intersect each other.

Input Format

The first line of input contains two positive integers, W and P - the number of wifi stations, and the number of points in the taxi trip, respectively.
Each of the next P lines contains two integers x and y - denoting the coordinates of the points in a trip, from starting point to destination point.
Then comes the description of W wifi stations. The first line of each Wifi station data contains an integer M, the number of vertices of that station. Each of the next M lines contains two integers x and y - denoting the coordinates of that station in either clockwise or counter-clockwise order.
For more info look at sample input / output.

Constraints

$1 \leq W \leq 32$
$2 \leq P \leq 32$
$3 \leq M \leq 32$
$-20{,}000 \leq xi, yi \leq 20{,}000$

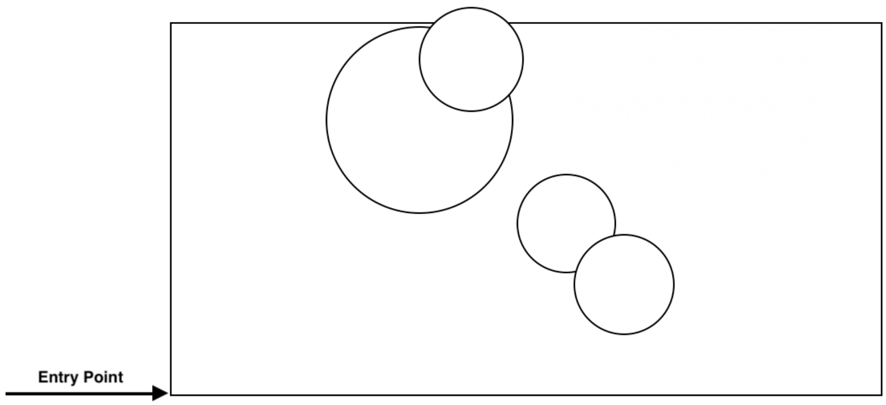
Output Format

Print one line of output containing the maximum distance the taxi gets from any wifi stations along it's path. The answer should be given with an absolute or relative error of at most 10^{-6}.

Sample Input	Sample Output
3 3 3 2 5 2 3 3 5 -3 1 -2 2 -1 1 -1 3 -3 3 4 0 1 0 2 1 2 1 1 4 2 1 4 1 4 4 2 4	1.000000

Problem G – Garden

Amir has a large garden full of apple trees. Every noon he walks into one of the trees and takes a nap under that tree, but he hates walking under the sunlight. His garden is rectangular and he has **N** trees in his garden. He enters the garden from the lower-left point of the garden (consider it as point (0,0), so coordinates of all the trees are greater than or equal to zero) and walks to the N-th tree. Each tree makes a shadow of radius R_i meters. Can you calculate the minimum distance Amir has to walk under the sunlight to get to the N-th tree?



Input Format

The first line of input contains an integer N, which denotes the number of trees. Each of the next N lines contains three space-separated integers, R_i, X_i, Y_i - radius of the shade of the i-th tree and its coordinates, respectively.

Constraints

$1 \leq N \leq 1000$

$0 < Ri \leq 100$

$0 \leq Xi, Yi \leq 100{,}000$

Output Format

Print one line of output containing the minimum distance Reza has to walk under the sunlight to rest under the N-th tree. The answer should be given with an absolute or relative error of at most 10^{-6} .

Sample Input	Sample Output
2 2 5 5 2 7 7	5.0710678

Problem H – Wifi

Masoud has just bought a new car. His car has a great acceleration unlike all the other cars in his city which has no acceleration at all, instead, those cars have Wifi, unlike Masoud's car. Masoud wants to travel along a road to get to X_t. At time 0 he starts at point $X = 0$ with an acceleration of $A_0\ m/s^2$ and speed of $V_0\ m/s$. At time 0 other cars are traveling along the road at a constant speed. All the people in this city are generous so all of the cars have free wifi. Whenever Masoud gets within a radius of 50 meters of each car, he can get connected to that car's wifi. Your task is to write a program that calculates the total distance that Masoud has not access to wifi. For the sake of simplicity assume cars are a single point which can surpass each other with no collision.

Input Format

The first line of input contains three space-separated integers V_0, A_0, X_t. The second line of input contains an integer N, the number of cars at constant speed. Each of the next N lines each contains two space-separated integers X_i and V_i, which denotes the initial position of that car and its speed.

Constraints

$0 \leq V_0, A_0 \leq 100\ and\ V_0 + A_0 > 0$

$0 < X_t \leq 10{,}000$

$50 < X_i \leq 10{,}000$

$0 \leq V_i \leq 100$

$0 \leq N \leq 1{,}000$

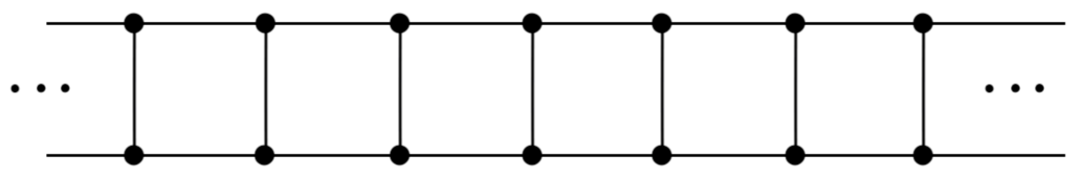
Output Format

Print one line of output, the **total distance** Masoud doesn't have access to free wifi. The answer should be given with an absolute or relative error of at most 10^{-6}.

Sample Input	Sample Output
10 1 200 2 60 2 80 4	37.51170

Problem I – Ladder

Lately, some computer scientists of Amirkabir University of Technology, including Reza, have designed an advanced robot that can help find injured people in a disaster, but somehow they want to use this robot in an exhibition to entertain people. Every August highschool students are invited to the university to get to know how AUT is like and why it is one of the best universities to study (except for the food and the security… etc). Each student is given a tree, a graph theory tree, (instead of a flower) as a souvenir from AUT. Some of these trees are magical and some are not, but nobody knows except that advanced robot. If the robot can put the tree on a ladder then the tree is magical and the student can study at AUT with no entrance exam. Each edge of the tree is of length 1, so is each segment of the ladder. Formally speaking your program must find a 1-1 matching between tree nodes and ladder intersections, so that if two nodes are connected in the tree, so are the corresponding two intersections on the ladder.



You can assume the robot has a ladder with infinite number of steps (like the above figure).

A few hours before the exhibition the software running the robot broke, and authorities have no idea what to do. Can you write a program that helps the robot to determine if a tree is magical or not?

Input Format

The first line of input contains an integer T - the number of test cases.
The first line of each test contains an integer n - the number of nodes in the tree.
Each of the next n-1 lines contains two space-separated integers. The nodes of the tree are numbered from 1 to n.

Constraints

$1 \leq T \leq 32$
$1 \leq n \leq 100{,}000$

Output Format

For each test print one line of output containing **Magical**, if that tree is Magical, otherwise print **NotMagical**.

Sample Input	Sample Output
7 1 2 2 3 2 4 4 5 5 6 6 7	Magical

Problem J – PaliT

Every year MeHdi receives an unusual birthday present! This year his friends have bought him a tree, not a garden tree, a tree in graph theory :(. Since he loves playing with trees, in his free time, he has written some letters on each edge of the tree. After he has no more edges to write letters on them, he wants to do something different, he wants to check whether the path between two nodes of the tree is palindrome or not? (a word or sequence that reads the same backwards as forwards, e.g. *madam* or *nursesrun*.)
Since he is busy with his other homework, he asks you to write a program which performs this task.

Input Format

The first line of input contains an integer n, the number of nodes in the tree.
The next n-1 lines contains x, y and label - two endpoints of an edge, and the corresponding letter on it.
The next line of input contains an integer q, the number of queries.
Each of the next q lines contains two integers, x and y - the two nodes which your program should determine whether the path between them is palindrome or not.

Constraints

$1 \leq n \leq 100{,}000$
$1 \leq q \leq 200{,}000$

Output Format

For each query print one line of output containing ***Palindrome*** if the path between x and y is a palindrome, otherwise print ***NotPalindrome***.

Sample Input	Sample Output
6 1 2 a 1 3 b 3 4 b 3 5 b 4 6 a 5 1 2 1 5 3 2 3 6 4 5	Palindrome Palindrome NotPalindrome NotPalindrome Palindrome

Problem K – Guardiola

Guardiola is the best football coach of all time. Each time he comes up with a new plan for training. He has drawn n points on the football field where point 1 and n are the two goals of the field (left goal and right goal). He has drawn some direct lines between some pairs of points which a player can dribble along it in both directions using the same amount of time.
He has designed two drills for his players in Manchester City. The first drill is dribbling. He puts a player on point 1 (let's call him player A) and asks him to dribble until he reaches point n. Now Guardiola wants to put some other players on the lines which he has drawn between points (as close to the points as possible, but not on the points) in a way that no matter which road player A chooses to dribble along for getting to point N, he has to dribble at least one other player on his path.
The second drill is practicing corner kicks. Some player will shoot a ball over point 1 or point n, the other players must get to that station as soon as possible.
Now Guardiola is facing this task, placing his players on the segments that he has drawn between points in such a way that each person dribbling from point 1 to n (or vice versa) will face at least one other player, and in case of a corner kick all players can get to point 1 or point n in the shortest possible time.

Input Format

The first line of input contains three space-separated integers, n, p and m - number of points on the ground, number of players Guardiola can use to put on lines, the number of lines between points.
Each of the next m lines contains three space-separated integers, **x**, **y** and **time**, which describes a segment between points **x** and **y** that takes **time** to traverse it.

Constraints

$1 \leq n \leq 200$
$1 \leq p \leq 50{,}000$
$1 \leq e \leq 5000$

Output Format

Print one line of output containing the shortest time all the players can get to point 1 or n with one decimal, or print **Impossible** if the number of players is not enough for drill one.

Sample Input	Sample Output
5 17 5 1 2 2 2 5 1 1 3 3 3 4 1 4 5 2	3.0