Problem A. Area

[Description]

To make sure that our company has the most immutable mobile phone all over the world, the Mokia decide to monitor all the cellphones on Earth. They have launched two satellites in order to track the information about hardness of the mobile phones. Although the satellites are able to scan all the mobile phones visible to them, unfortunately, with only two satellites, our Mokia can't cover all surface on Earth simultaneously. One satellite can only monitor the surface which is unblock to its current location. The satellite can only be blocked by the Earth itself, that is to say, one point of the Earth's surface is visible to one satellite if and only if the line between this point and the satellite does not intersect with Earth. No relativistic effects are considered.

The CEO of Mokia, Steve, wants to get the current coverage ratio of the satellites. Since we only know the current location of the satellites, it is very hard to know the exact number of ratio. To simplify this task, we can consider Earth as a perfect sphere of radius R. With this simplification, and the locations of the satellites, this task is pretty easy.

[Input]

The input file leads with one line contains the number of test cases T. (T <= 1000)
For each test case, the first line contains only one real number R (0.01 <= R <= 10000), indicating the radius of Earth.
The second line and third line each contains three real numbers, indicating the current location of the two satellites (0.01 <= X, Y, Z <= 10000). And you may assume the origin of coordinate system is located on the center of Earth, and two satellites are above the surface of Earth.
Also, all real numbers have at most two digits after the decimal point.

[Output]

For each test cases, output only one line with the test number and the coverage of the two satellites. All coverage should be presented as percentages and rounded to five decimal places.

[Sample Input]

```
2
1
0 0 2
0 0 -2
1
0 0 2
0 0 -2.5
```

[Sample Output]

```
Case #1: 50.00000%
Case #2: 55.00000%
```

Problem B. Parade Show

[Description]

2013 is the 60 anniversary of Nanjing University of Science and Technology, and today happens to be the anniversary date. On this happy festival, school authority hopes that the new students to be trained for the parade show. You should plan a better solution to arrange the students by choosing some queues from them preparing the parade show. (one student only in one queue or not be chosen)

Every student has its own number, from 1 to n. (1<=n<=10^5), and they are standing from 1 to n in the increasing order the same with their number order. According to requirement of school authority, every queue is consisted of exactly m students. Because students who stand adjacent in training are assigned consecutive number, for better arrangement, you will choose in students with in consecutive numbers. When you choose these m students, you will rearrange their numbers from 1 to m, in the same order with their initial one.

If we divide our students' heights into k (1<=k<=25) level, experience says that there will exist an best viewing module, represented by an array a[]. a[i] (1<=i<=m)stands for the student' s height with number i. In fact, inside a queue, for every number pair i, j (1<=i,j<=m), if the relative bigger or smaller or equal to relationship between the height of student number i and the height of student number j is the same with that between a[i] and a[j], then the queue is well designed. Given n students' height array x[] (1<=x[i]<=k), and the best viewing module array a[], how many well designed queues can we make at most?

[Input]

Multiple cases, end with EOF.
First line, 3 integers, n (1<=n<=10ˆ5) m (1<=m<=n) k(1<=k<=25),
Second line, n students' height array x[] (1<=x[i]<=k,1<=i<=n);
Third line, m integers, best viewing module array a[] (1<=a[i]<=k,1<=i<=m);

[Output]

One integer, the maximal amount of well designed queues.

[Sample Input]

```
10 5 10
2 4 2 4 2 4 2 4 2 4
1 2 1 2 1
```

[Sample Output]

```
1
```

Problem C. Count The Pairs

[Description]

With the 60th anniversary celebration of Nanjing University of Science and Technology coming soon, the university sets n tourist spots to welcome guests. Of course, Redwood forests in our university and its Orychophragmus violaceus must be recommended as top ten tourist spots, probably the best of all. Some undirected roads are made to connect pairs of tourist spots. For example, from Redwood forests (suppose it’s a) to fountain plaza (suppose it’s b), there may exist an undirected road with its length c. By the way, there is m roads totally here. Accidently, these roads’ length is an integer, and all of them are different. Some of these spots can reach directly or indirectly to some other spots. For guests, they are travelling from tourist spot s to tourist spot t, they can achieve some value f. According to the statistics calculated and recorded by us in last years, We found a strange way to calculate the value f:

From s to t, there may exist lots of different paths, guests will try every one of them. One particular path is consisted of some undirected roads. When they are travelling in this path, they will try to remember the value of longest road in this path. In the end, guests will remember too many longest roads’ value, so he cannot catch them all. But, one thing which guests will keep it in mind is that the minimal number of all these longest values. And value f is exactly the same with the minimal number.

Tom200 will recommend pairs (s, t) (start spot, end spot points pair) to guests. P guests will come to visit our university, and every one of them has a requirement for value f, satisfying f>=t. Tom200 needs your help. For each requirement, how many pairs (s, t) you can offer?

[Input]

Multiple cases, end with EOF.
First line:n m
n tourist spots (1<n<=10000), spots’ index starts from 0.
m undirected roads (1<m<=500000).

Next m lines, 3 integers, a b c
From tourist spot a to tourist spot b, its length is c. 0<a, b<n, c(0<c<1000000000), all c are different.

Next one line, 1 integer, p (0<p<=100000)
It means p guests coming.

Next p line, each line one integer, t(0<=t)
The value t you need to consider to satisfy f>=t.

[Output]

For each guest's requirement value t, output the number of pairs satisfying f>=t.
Notice, (1,2), (2,1) are different pairs.

[Sample Input]

```
2 1
0 1 2
3
1
2
3
3 3
0 1 2
0 2 4
```

```
1 2 5
5
0
2
3
4
5
```

[Sample Output]

```
2
2
0
6
6
4
4
0
```

Problem D. Divide Groups

[Description]

This year is the 60th anniversary of NJUST, and to make the celebration more colorful, Tom200 is going to invite distinguished alumnus back to visit and take photos.

After carefully planning, Tom200 announced his activity plan, one that contains two characters:

1. Whether the effect of the event are good or bad has nothing to do with the number of people join in.

2. The less people joining in one activity know each other, the more interesting the activity will be. Therefore, the best state is that, one dose not know anyone except himself.

The event appeals to a great number of alumnus, and Tom200 finds that they may not know each other or may just unilaterally recognize others. To improve the activities effects, Tom200 has to divide all those who signed up into groups to take part in the activity at different time. As we know, one's energy is limited, and Tom200 can hold activity twice. Tom200 already knows the relationship of each two person, but he cannot divide them because the number is too large.

Now Tom200 turns to you for help. Given the information, can you tell if it is possible to complete the dividing mission to make the two activity in best state.

[Input]

The input contains several test cases, terminated by EOF.

Each case starts with a positive integer n (2<=n<=100), which means the number of people joining in the event.

N lines follow. The i-th line contains some integers which are the id

of students that the i-th student knows, terminated by 0. And the id starts from 1.

[Output]

If divided successfully, please output "YES" in a line, else output "NO".

[Sample Input]

```
3
3 0
1 0
1 2 0
```

[Sample Output]

```
YES
```

Problem E. Polygon

[Description]

In this problem, you're given a function f(x) = ax^2 + bx + c(x in [l, r]) and a simple polygon. And we need you to figure out the length of f(x) in this polygon.

[Input]

There are several test cases and the cases end with EOF. For each case:

The first line contains six integers n (1 <= n <= 20000), a(a != 0), b, c, l and r, which are the total of the points of the polygon and the parameters of the function and these parameters are in the range [-20000, 20000].

The next n lines contains two integers x and y, which are the coordinate of the point and both of them are in the range[-10^9, 10^9]. We can ensure that there is a edge between i-th point and (i+1)-th point (1 <= i < n) and there is a edge between first point and n-th point.

[Output]

For each case, output your answer, your answer should rounded to two decimal places.

[Sample Input]

```
4 0 0 0 -100 100
-10 0
10 0
10 10
-10 10
```

```
4 1 0 0 -100 100
-10 0
10 0
10 10
-10 10
```

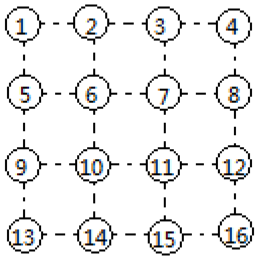
[Sample Output]

```
20.00
21.52
```

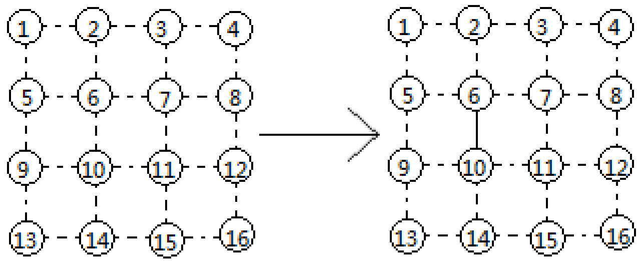
Problem F. Fishhead’ s Little Game

[Description]

There is a 3 by 3 grid and each vertex is assigned a number.

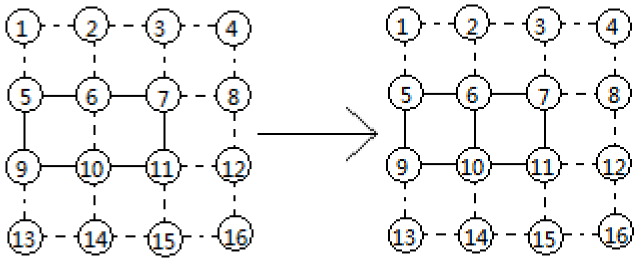


It looks like JiuGongGe, but they are different, for we are not going to fill the cell but the edge. For instance,



adding edge 6 -> 10

The rule of this game is that each player takes turns to add an edge. You will get one point if the edge you just added, together with edges already added before, forms a new square (only square of size 1 is considered). Of course, you get two points if that edge forms two squares. Notice that an edge can be added only once.



forming two squares to get two points

Tom200 and Jerry404 is playing this little game, and have played n rounds when Fishhead comes in. Fishhead wants to know who will be the winner. Can you help him? Assume that Tom200 and Jerry404 are clever enough to make optimal decisions in each round. Every Game starts from Tom200.

[Input]

The first line of the input contains a single integer T (T <= 100), the number of test cases.

For each case, the first line contains an integers n (12 <= n <= 24), which means they have taken total n rounds in turn. Next n lines each contains two integers a, b (a, b <= 16) representing the two endpoints of the edge.

[Output]

For each case output one line "Case #X: ", representing the Xth case, starting from 1. If Tom200 wins, print "Tom200" on one line, print "Jerry404" otherwise.

[Sample Input]

```
1
15
1 2
1 5
2 6
5 9
6 10
9 10
5 6
2 3
3 7
7 11
10 11
3 4
6 7
7 8
4 8
```

[Sample Output]

```
Case #1: Tom200
```

[Hint]

In case 1, Tom200 gets two points when she add edge 5 -> 6, two points in edge 6 -> 7, one point in 4 -> 8.

Problem G. Spacecraft Monitoring

[Description]

In recent years, China's space industry has made rapid development, and recently launched the manned spacecraft shenzhou 10. Beauty Astronaut Wang Yaping even made a lesson in the spacecraft to the whole world. It means a lot to China's space communication. And the Nanjing University of Science and Technology also contribute their strength in communication, radar and detecting guidance.

Today, there have been many aircrafts in space, they are communicating with each other in air, or with the ground monitoring points in ground (including fixed and mobile monitoring point). To simplify the problem, we assume that all the aircrafts and ground monitoring points are all in the equatorial plane. Aircrafts are all in synchronous orbit, suppose the earth' s core as point (0, 0). We already know:

Every aircraft' s or ground monitoring point' s height R relative to the earth' s core, initial angle A (rad), and some communication relationship between them.

Also every aircraft' s or ground monitoring point' s cost price P.

And the ground-to-ground price per unit distance: dd, ground-to-air price per unit distance: dk (air-to-ground the same), air-to-air price per unit distance: kk.

In order to know the general condition of the system, you should calculate the maximal average cost price of some aircrafts or ground monitoring points. The average cost price = (the sum of aircrafts' or ground monitoring points' cost price + communication cost price between them if it's existing)/(the amount of the aircrafts or ground monitoring points)

[Input]

Multiple cases, end with EOF;

The first line, 4 positive integers, n, dd, dk, kk, there is n aircrafts or ground monitoring points, dd, dk, kk is the same with description above. Remember that we first calculate the distance between two objects into integer, four to five homes in rounding, then calculate the communication cost price.

Next n lines, each represents an aircraft or ground monitoring point,

if the first letter is 'd', it means "ground monitoring point" in ground,
if the first letter is 'k', it means "aircraft" in air,
then R,A,P, described above.

Next an integer m, the amount of communication relationship.

Next m lines, each line is like x, B, y, C (1<=B, C<=n, x, y = 'd' or 'k'), represents object B is communicated with object C, B's type is x, C's type is y; B,C is the sequence number in particular kind ('d' or 'k') listed above, starts from 1.

1<=n<=300, 0<=m<=n*n, 1<=dd,dk,kk<=10^6, R、A are floating numbers, 0<=R<=1000,
0<=A<2*PI, 1<=P<=10^6。

PS: communication between two points is considered as a segment.

[Output]

After "Case #" + case number + ": ", case number starts from 1, Output 2 numbers, separated by a blank space, first is the maximal average cost price, next is the amount of this part. The result should be corrected to four decimal places.

[Sample Input]

```
2 1000 2000 3000
d 2 0 1
d 3 0 4
1
d 1 d 2
```

[Sample Output]

```
Case #1: 502.5000 2
```

Problem H. Heroes of Might and Magic

[Description]

《Heroes of Might and Magic III: The Restoration of Erathia》 is one of the most classic SLG game in recent years. Its powerful strategies and tactics to attract the players is enduring. Everyone controls his own combat arms to battle. Decide the outcome of a battle, in addition to a hero’s magic, level, character action good or bad, the layout of soldiers to walk, the most important thing is one’s own quantity and quality of all kinds of arms.

Jerry404 likes this game so much and later, in order to improve efficiency, his favorite is the automatic battle function inside.

Tom200 knows that and wonders what algorithm the system is through to calculate its automatically fight.

After a long time of YY, Tom200 finally speculated the automatic fighting rules:

1. Each hero can take up to seven arms, and the hero’s attack power, defense power bonus to each of the upper arms. For example, the hero’s attack power and defense power are respectively Ha and Hd, and the arms’ original attack

power and defense power are Aa and Ad. After addition, Aa = Aa+Ha , Ad =Ad+Hd, and Aa is greater than 0.

2. Hero's magic value equals its intelligence number * 10.
Hero's magic injury equals its strength number * 50.
Heroes' single cast magic consumption is fixed at 12, during the battle the hero cannot be attacked.

3. Each arms have five attributes: attack power, defense power, the value of life, the number of units and attack speed. Assume that A arms attack B arms, A arms of the attribute were Aa, Ad, Ab, Am, As and correspondingly B arms' five attributes were Ba, Bd, Bb, Bm, Bs. Therefore, the total injury of A arms equals Am*Aa. At that time, B arms will bear that hurt unit by unit:

if Am*Aa is not greater than Bd, then one unit of B arms will deduct one value of life;

if Am*Aa is greater than Bd, then the excess part of the attack force will be considered as the deduction of current unit's value of life of B arms;

if one unit's current life of value cannot afford, then it dies and we will take the remainder of the attack force into the next unit settlement. Also at this time, you should calculate if attack power is greater than defense power.

if all units of the current arms are dead, injury will no longer be calculated.

For example, assume Aa = 5, Am = 2, Bd = 3, Bb = 4, Bm = 2, and A's total injury is 10, greater than Bd by 7, and Bb is 4 which cannot afford. Then one unit dies and the left attack power is 3, not greater than Bd, so the second unit deduct one value of life. Finally, the end result is that B arms left 1 unit whose life is 3.

4. For brevity, the arms in left and right attack each other in turn every round (the arms in the left attacks first. Hero A is always on the left, B right, each round can only attack once per arms, if the current round of a party has no arms to attack, then the side bye). Attack speed determines the arms of the side order of attack and the arms appear first move first in case of the same attack speed. Heroes always turn to attack at the beginning of each round and its attack effects can be calculated in the same way as (3) describes. In every round, the hero cannot attack if it has no magic value and in this case it is seen as an attack without injury. A single round ends when heroes and arms of both sides attacked. The fourth rule can also look at the following schematic attack per round:

A hero magic attack -> B hero magic attack -> A fastest attack speed arms that are alive -> B fastest attack speed arms that are alive -> A second fastest attack speed arms that are alive -> B fastest attack speed arms that are alive...

5. Assuming that each arms' attack range if full screen and can be attacked by every other arms.

6. The hero can equip treatment tent, which can get back 20 points of life value in each round. The treatment tent will choose the arms in which an individual unit has the maximum difference between remaining life value and original life value. In case of the same maximum difference, the tent will choose the arms that appear first (The amount of one unit's life value will not exceed its original value and the tent cannot bring back those who are already dead.)

7. Every round all the arms (including the hero) will attack the enemy arms which has the most number of units. In case of the same number, it will attack the arms that appear first.

8. Every round each arms will have a chance of counter attack and it will counter attack immediately if it is able to. (If attacked by a hero, it cannot counter attack.)

9. A round ends after all the arms' movement.

10. All the numbers are integers and the single number value is no greater than 100 and no less than 0. Calculations are integer calculations (rounded down is all right.)

11. The game comes to end when every single unit of all the arms on one side are dead.

Tom200 wants you to tell the final ending according to these crazy rules.

[Input]

The first line contains a number T denoting the number of the test cases.

The fisrt line of each case are six integers Aa,Ad,Ae,As,An(1<=An<=7),Azp, indicating one's own hero's attack power, defense power, intelligence, strength, the number of arms the hero takes.

If Azp=1 ,it means there is a treatment tent; else if Azp=0, there is no tent.

Five lines follow and each line contains An integers , respectively indicating An arms' attack power, defense power, the value of life, the number of units and attack speed.

The next line is the corresponding data of the enemy hero :Ba,Bd,Be,Bs,Bn,Bzp.

Five lines follow and each line contains Bn integers , respectively indicating Bn arms' attack power, defense power, the value of life, the number of units and attack speed.

[Output]

If one's own side (A hero) wins, output "Win!" in a line, else output "Lose!".

The next two lines output the number of dead units of each arms in order of arms appearance.

We ensure that one side's arms will all die, in other words endless draw does not exist.

[Sample Input]

```
1
1 1 4 3 5 1
5 4 3 2 1
3 2 1 0 0
4 3 2 2 1
2 3 4 5 6
5 3 2 1 1
2 1 2 1 4 0
3 3 2 4
4 3 5 2
2 1 1 2
4 3 2 1
3 2 4 5
```

[Sample Output]

```
Win!
0 0 0 2 6
4 3 2 1
```

Problem I. Install Air Conditioning

[Description]

NJUST carries on the tradition of HaJunGong. NJUST, who keeps up the " people-oriented, harmonious development” of the educational philosophy and develops the ” unity, dedication, truth-seeking, innovation” school motto, has now become an engineering-based, multidisciplinary university.

As we all know, Nanjing is one of the four hottest cities in China. Students in NJUST find it hard to fall asleep during hot summer every year. They will never, however, suffer from that hot this year, which makes them really excited. NJUST’ s 60th birthday is approaching, in the meantime, 50 million is spent to install air conditioning among students dormitories. Due to NJUST’ s long history, the old circuits are not capable to carry heavy load, so it is necessary to set new high-load wires. To reduce cost, every wire between two dormitory is considered a segment. Now, known about all the location of dormitories and a power plant, and the cost of high-load wire per meter, Tom200 wants to know in advance, under the premise of all dormitories being able to supply electricity, the minimum cost be spent on high-load wires. And this is the minimum strategy. But

Tom200 is informed that there are so many wires between two specific dormitories that we cannot set a new high-load wire between these two, otherwise it may have potential risks. The problem is that Tom200 doesn't know exactly which two dormitories until the setting process is started. So according to the minimum strategy described above, how much cost at most you'll spend?

[Input]

The first line of the input contains a single integer T(T ≤ 100), the number of test cases.

For each case, the first line contains two integers n(3 ≤ n ≤ 1000), k(1 ≤ k ≤ 100). n represents n - 1 dormitories and one power plant, k represents the cost of high-load wire per meter. n lines followed contains two integers x, y(0 ≤ x, y ≤ 10000000), representing the location of dormitory or power plant. Assume no two locations are the same, and no three locations are on a straight line. The first one is always the location of the power plant.

[Output]

For each case, output the cost, correct to two decimal places.

[Sample Input]

```
2

4 2
0 0
1 1
2 0
3 1

4 3
0 0
1 1
1 0
0 1
```

[Sample Output]

```
9.66
9.00
```

Problem J. Tree

[Description]

Zero and One are good friends who always have fun with each other. This time, they decide to do something on a tree which is a kind of graph that there is only one path from node to node. First, Zero will give One an tree and every node in this tree has a value. Then, Zero will ask One a series of queries. Each query contains three parameters: x, y, z which mean that he want to know the maximum value produced by z xor each value on the path from node x to node y (include node x, node y). Unfortunately, One has no idea in this question. So he need you to solve it.

[Input]

There are several test cases and the cases end with EOF. For each case:

The first line contains two integers n(1<=n<=10ˆ5) and m(1<=m<=10ˆ5), which are the amount of tree’ s nodes and queries, respectively.

The second line contains n integers a[1..n] and a[i](0<=a[i]<2ˆ{16}) is the value on the ith node.

The next n－1 lines contains two integers u v, which means there is an connection between u and v.

The next m lines contains three integers x y z, which are the parameters of Zero’ s query.

[Output]

For each query, output the answer.

[Sample Input]

```
3 2
1 2 2
1 2
2 3
1 3 1
2 3 2
```

[Sample Output]

```
3
0
```

Problem K. Walk Through Squares

[Description]

On the beaming day of 60th anniversary of NJUST, as a military college which was Second Artillery Academy of Harbin Military Engineering Institute before, queue phalanx is a special landscape.

Here is a M*N rectangle, and this one can be divided into M*N squares which are of the same size. As shown in the figure below:

```
01--02--03--04
||  ||  ||  ||
05--06--07--08
||  ||  ||  ||
09--10--11--12
```

Consequently, we have (M+1)*(N+1) nodes, which are all connected to their adjacent nodes. And actual queue phalanx will go along the edges.

The ID of the first node, the one in top-left corner, is 1. And the ID increases line by line first ,and then by column in turn ,as shown in the figure above.

For every node, there are two viable paths:

(1)go downward, indicated by 'D';

(2)go right, indicated by 'R';

The current mission is that, each queue phalanx has to walk from the left-top node No.1 to the right-bottom node whose id is (M+1)*(N+1).

In order to make a more aesthetic marching, each queue phalanx has to conduct two necessary actions. Let's define the action:

An action is started from a node to go for a specified travel mode. So, two actions must show up in the way from 1 to (M+1)*(N+1).

For example, as to a 3*2 rectangle, figure below:

```
 01--02--03--04
||  ||  ||  ||
05--06--07--08
||  ||  ||  ||
09--10--11--12
```

Assume that the two actions are (1)RRD (2)DDR

As a result , there is only one way : RRDDR. Briefly, you can not find another sequence containing these two strings at the same time.

If given the N, M and two actions, can you calculate the total ways of walking from node No.1 to the right-bottom node ?

[Input]

The first line contains a number T,(T is about 100, including 90 small test cases and 10 large ones) denoting the number of the test cases.

For each test cases,the first line contains two positive integers M and N(For large test cases,1<=M,N<=100, and for small ones 1<=M,N<=40). M denotes the row number and N denotes the column number.

The next two lines each contains a string which contains only 'R' and 'D'. The length of string will not exceed 100. We ensure there are no empty strings and the two strings are different.

[Output]

For each test cases,print the answer MOD 1000000007 in one line.

[Sample Input]

```
2
3 2
RRD
DDR
3 2
R
D
```

[Sample Output]

```
1
10
```