acm International Collegiate Programming Contest

2013 ACM-ICPC China Hunan Invitational Programming Contest

Problem Set

May 26, 2013

Hosted by Hunan University

Changsha, China

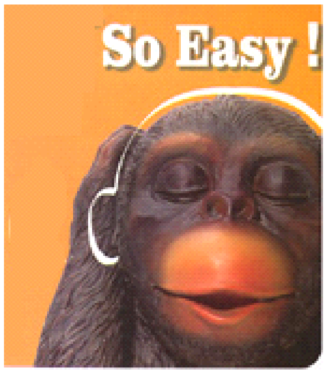
Problem Set

Problem A

So Easy!

Description

A sequence S_n is defined as:



$$S_n = \left\lceil \left(a+\sqrt{b}\right)^n \right\rceil \% m$$

Where a, b, n, m are positive integers. $\lceil x \rceil$ is the ceil of x. For example, $\lceil 3.14 \rceil = 4$. You are to calculate S_n.

You, a top coder, say: So easy!

Input

There are several test cases, each test case in one line contains four positive integers: a, b, n, m. Where $0< a, m < 2^{15}$, $(a-1)^2< b < a^2$, $0 < b, n < 2^{31}$.
The input will finish with the end of file.

Output

For each the case, output an integer S_n.

Sample Input

```
2 3 1 2013
2 3 2 2013
2 2 1 2013
```

Sample Output

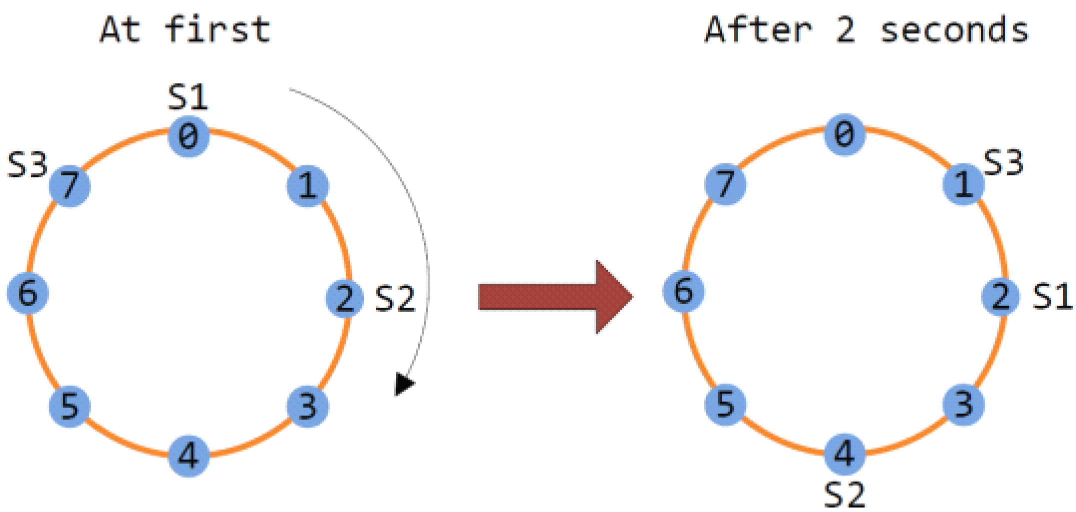
```
4
14
4
```

Problem B

Circle Game

Description

You must have some knowledge with circle games in ACM problems, such as Joseph Ring problem. Today we are going to introduce a new circle game described as follows. There is a circle which contains ***M*** points, numbered from 0 to *M*-1. At first, ***n*** students stand in different points of the circle. It is possible that more than one student stand in the same point. For each second, each student moves one step forward in clockwise direction. The following picture gives us an example of that *M* equals to 8 and *n* equals to 3. Initially, the first student S1 stands in the point 0, the second student S2 stands in the point 2 and the third student stands in the point 7. After 2 seconds, they will change their positions as the right part of the picture.



At the beginning of the game, Tracy writes down the positions of all students. Then he will go to sleep and students will abide by the above rules to make this game run until Tracy wake up. After a period of time, Tracy wakes up. He finds it is very hard to identify where the first student S1 stands in and where the second student S2 stands in and so on, because all students are the exactly similar and unable to be distinguished. So Tracy records all the positions of students and writes down the XOR sum of these positions (The XOR sum of an array A means A[0] xor A[1] xor … A[*n*-2] xor A[*n*-1] and xor is the exclusive or operation, for example 0111 xor 1011 = 1100). Now Tracy wants you to help him to know how many seconds he had slept. To simplify the problem, the number of points is always a power of 2 like $\mathbf{2^m}$.

Furthermore, Tracy knows that the time he slept is not greater than ***T***. Please note that the time Tracy slept must be greater than zero. There may have many different periods of time corresponding with the above conditions.

Input

There are several test cases, each test case firstly contains four integers ***n***, ***m***, ***S***, ***T***. *S* is the XOR sum of students' positions when Tracy wakes up. Another 3 integers *n*, *m*, *T* are described as the above. Then *n* integers will follow in the next line, which represent the positions of students initially. The input will finish with the end of file. $0 < n <= 100000$, $0 < m <= 50$, $0 <= S < 2^{50}$, $0 < T <= 10^{16}$ and the student position is in the range $[0, 2^m)$.

Output

For each test case, output the number of possible periods of time that Tracy had slept and matched the above restrictions. Please output zero if no time matched.

Sample Input

```
3 3 7 5
0 2 7
5 3 7 5
1 2 3 4 5
4 4 0 10
1 3 5 7
6 5 18 100
22 10 18 20 2 10
```

Sample Output

```
1
0
4
50
```

Hint

For the first test case, the following table explains that only "2 seconds" matched the restrictions.

Time (seconds)	Positions after a period of time	XOR sum
1	1 3 0	2
2	**2 4 1**	**7**
3	3 5 2	4
4	4 6 3	1
5	5 7 4	6

Problem C

Brilliant Programmers Show

Description

Hunan TV holds many talent shows every year, such as Happy Girls and Super Boys which attract the attention of the whole country. This year Hunan University held a new type of talent show called Brilliant Programmers. Millions of programmers had registered online and only top N most brilliant programmers got the opportunity to compete on site. The organizer had hold ten rounds of qualification contest and programmers were ranked by their total scores. The programmers who ranked after N were eliminated.

The final show continued for a very long time. Initially programmers were ranked by their qualification scores. The rule was special: A challenge may be happen between exactly two adjacent ranked programmers at any time and the lower ranked one tries to solve the other's problem. If the challenger successfully solves this problem, their ranks exchange. Otherwise their ranks remain unchanged. It is guaranteed that a programmer never involved in two challenges at the same time. The top ranked programmer at last is the champion.

The show was over but… The hard disk which logs the whole progress was burned out. After data rescue, the number of successful challenges of each programmer was recovered but the final rank was disappeared forever. During the rescue some errors may occur, which lead to some wrong recovered numbers. Is the show possible at all? If it is possible, can you help to find the champion from the very limited information?

Input

There are multiple test cases.

Each test case is described in two lines. The first line contains one integer N: the number of programmers. The second line contains a sequence of integers Ai that gives the number of successful challenges of the programmer initially ranked i-th.

$1 <= N <= 10^6$, $0 <= Ai <= 10^9$

The input will finish with the end of file.

Output

For each case the output contains only one line.

If it is an impossible show, output "Bad Rescue". Otherwise if the champion is uniquely determined, output the initial rank of the champion. Output "Unknown" if the champion is not sure.

Sample Input

2
0 1
3
0 1 5
3
0 1 1

Sample Output

2
Bad Rescue
Unknown

Problem D

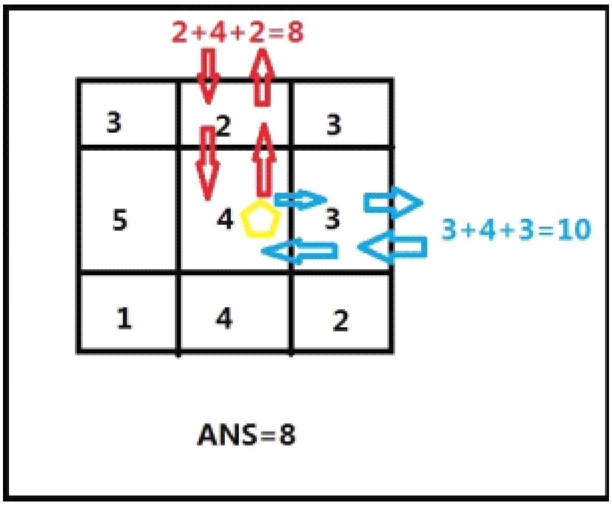
Hunter

Description

One day, a hunter named James went to a mysterious area to find the treasures. James wanted to research the area and brought all treasures that he could.

The area can be represented as a $N*M$ rectangle. Any points of the rectangle is a number means the cost of research it,-1 means James can't cross it, James can start at any place out of the rectangle, and explore point next by next. He will move in the rectangle and bring out all treasures he can take. Of course, he will end at any border to go out of rectangle(James will research every point at anytime he cross because he can't remember whether the point are researched or not).

Now give you a map of the area, you must calculate the least cost that James bring out all treasures he can take(one point up to only one treasure).Also, if nothing James can get, please output 0.



Input

The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case begins with a line containing 2 integers N M, ($1<=N,M<=200$), that represents the rectangle. Each of the following N lines contains M numbers(0~9),represent the cost of each point. Next is K($1<=K<=13$),and next K lines, each line contains 2 integers x y means the position of the treasures, x means row and start from 0, y means column start from 0 too.

Output

For each test case, you should output only a number means the minimum cost.

Sample Input

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

Sample Output

```
8
11
```

Problem E

Special equations

Description

Let $f(x) = a_n x^n + ... + a_1 x + a_0$, in which a_i ($0 <= i <= n$) are all known integers. We call $f(x) \equiv 0 \pmod{m}$ congruence equation. If m is a composite, we can factor m into powers of primes and solve every such single equation after which we merge them using the Chinese Reminder Theorem. In this problem, you are asked to solve a much simpler version of such equations, with m to be prime's square.

Input

The first line is the number of equations T, $T<=50$.

Then comes T lines, each line starts with an integer deg ($1<=deg<=4$), meaning that f(x)'s degree is deg. Then follows deg integers, representing a_n to a_0 ($0 < abs(a_n) <= 100$; $abs(a_i) <= 10000$ when $deg >= 3$, otherwise $abs(a_i) <= 100000000$, $i<n$). The last integer is prime pri ($pri<=10000$).

Remember, your task is to solve $f(x) \equiv 0 \pmod{pri*pri}$

Output

For each equation $f(x) \equiv 0 \pmod{pri*pri}$, first output the case number, then output anyone of x if there are many x fitting the equation, else output "No solution!"

Sample Input

```
4
2 1 1 -5 7
1 5 -2995 9929
2 1 -96255532 8930 9811
4 14 5458 7754 4946 -2210 9601
```

Sample Output

```
Case #1: No solution!
Case #2: 599
Case #3: 96255626
Case #4: No solution!
```

Problem F

Multi-bit Trie

Description

IP lookup is one of the key functions of routers for packets forwarding and classifying. Generally, IP lookup can be simplified as a Longest Prefix Matching (LPM) problem. That's to find the longest prefix in the Forwarding Information Base (FIB) that matches the input packet's destination address, and then output the corresponding Next Hop information.

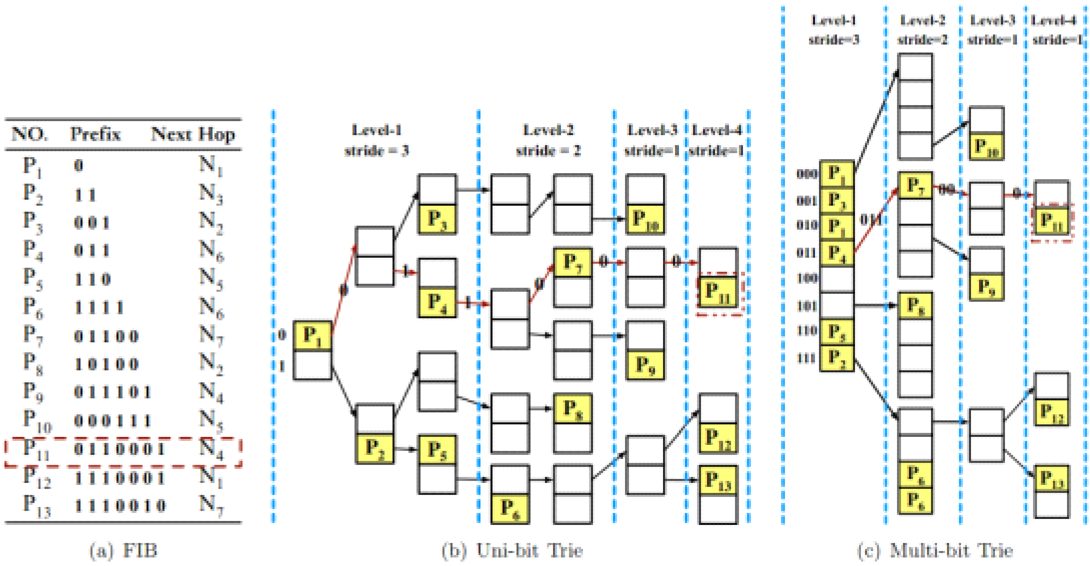


Figure 1: A sample of Uni-bit Trie and its corresponding Multi-bit Trie.

Trie-based solution is the most wildly used one to solve LPM. As shown in Fig.1(b), an uni-bit trie is just a binary tree. Processing LPM on it needs only traversing it from the root to some leaf, according to the input packet's destination address. The longest prefix along this traversing path is the matched one. In order to reduce the memory accesses for one lookup, we can compress some consecutively levels of the Uni-bit Trie into one level, transforming the Uni-bit Trie into a Multi-bit Trie.

For example, suppose the strides array is {3, 2, 1, 1}, then we can transform the Uni-bit Trie shown in Fig.1(b) into a Multi-bit Trie as shown in Fig.1(c). During the transforming process, some prefixes must be expanded. Such as 11(P2), since the first stride is 3, it should be expanded to 110(P2) and 111(P2). But 110(P5) is already exist in the FIB, so we only store the longer one 110(P5).

Multi-bit Trie can obviously reduce the tree level, but the problem is how to build a Multi-bit Trie with the minimal memory consumption (the number of memory units). As shown in Fig.1, the Uni-bit Trie has 23 nodes and consumes 46 memory units in total, while the Multi-bit Trie has 12 nodes and consumes 38 memory units in total.

Input

The first line is an integer *T*, which is the number of testing cases.

The first line of each case contains one integer L, which means the number of levels in the Uni-bit Trie.

Following L lines indicate the nodes in each level of the Uni-bit Trie.

Since only 64 bits of an IPv6 address is used for forwarding, a Uni-bit Trie has maximal 64 levels. Moreover, we suppose that the stride for each level of a Multi-bit Trie must be less than or equal to 20.

Output

Output the minimal possible memory units consumed by the corresponding Multi-bit Trie.

Sample Input

```
1
7
1
2
4
4
5
4
3
```

Sample Output

```
38
```

Problem G

Travel in time

Description

Bob gets tired of playing games, leaves Alice, and travels to Changsha alone. Yuelu Mountain, Orange Island, Window of the World, the Provincial Museum etc...are scenic spots Bob wants to visit. However, his time is very limited, he can't visit them all.

Assuming that there are N scenic spots in Changsha, Bob defines a satisfaction value Si to each spot. If he visits this spot, his total satisfaction value will plus Si. Bob hopes that within the limited time T, he can start at spot S, visit some spots selectively, and finally stop at spot E, so that the total satisfaction value can be as large as possible. It's obvious that visiting the spot will also cost some time, suppose that it takes Ci units of time to visit spot i ($0 <= i < N$).

Always remember, Bob can choose to pass by a spot without visiting it (including S and E), maybe he just want to walk shorter distance for saving time.

Bob also has a special need which is that he will only visit the spot whose satisfaction value is **strictly** larger than that of which he visited last time. For example, if he has visited a spot whose satisfaction value is 50, he would only visit spot whose satisfaction value is 51 or more then. The paths between the spots are bi-directional, of course.

Input

The first line is an integer W, which is the number of testing cases, and the W sets of data are following.

The first line of each test data contains five integers: N M T S E. N represents the number of spots, $1 < N < 100$; M represents the number of paths, $0 < M < 1000$; T represents the time limitation, $0 < T <= 300$; S means the spot Bob starts from. E indicates the end spot. ($0 <= S, E < N$)

The second line of the test data contains N integers Ci ($0 <= Ci <= T$), which means the cost of time if Bob visits the spot i.

The third line also has N integers, which means the satisfaction value Si that can be obtained by visiting the spot i ($0 <= Si < 100$).

The next M lines, each line contains three integers u v L, means there is a bi-directional path between spot u and v and it takes L units of time to walk from u to v or from v to u. ($0 <= u, v < N, 0 <= L <= T$)

Output

Output case number in the first line (formatted as the sample output).

The second line contains an integer, which is the greatest satisfaction value. If Bob can't reach spot E in T units of time, you should output just a "0" (without quotation marks).

Sample Input

```
1
4 4 22 0 3
1 1 1 1
5 7 9 12
0 1 10
1 3 10
0 2 10
2 3 10
```

Sample Output

```
Case #1:
21
```

Problem H

Bottles Arrangement

Description

Hunan cuisine is really wonderful! But if you don't like spicy food, you will feel terrible since it can be hard for you to find any food without hot pepper here. Big Fan is a student from the north who was not fit to the spicy food in Changsha. He became thinner and thinner because eating little food and maintained his life mostly by drinking water. One day, he found that the wine in Hunan is pretty good, such as Jiugui, Liuyang River, Shaoyang Daqu and so on. He got addicted to it and became an alcoholic, leading a depressed life.

Now N days have passed and he is sobered. He is surprised to find that there are exactly $N\times M$ bottles around him. Another amazing fact is that there are N bottles with height 1 and N bottles with height 2 … N bottles with height M.

Now he is interested in playing with these bottles. He wants to arrange all these bottles in a rectangle with M rows and N columns which satisfied:

- In any column, there are no bottles with same height;
- In any row, the height difference between any two adjacent bottles is no more than 1.

He defined a strange function Y which equals the maximum value of the total height of any single row. He is addicted in arranging these rubbish bottles to find the minimal Y. You know that he cannot solve it with his pour IQ. You are his friend and can't endure his decadence any more. So you decide to help him solve this problem and then bring him back to study.

Input

There are several test cases. For each case, the input contains one line with two integers M and N ($1< M <= 10000$, $3 <= N < 2\times M$, It is guaranteed that N is always odd).

The input will finish with the end of file.

Output

For each test case, print the minimal Y in single line.

Sample Input

3 3
3 5

Sample Output

8
11

Hint

For the first case the solution is:
1 2 3
2 1 1
3 3 2

Problem I

Throw the Stones

Description

Remember our childhood? A few naked children throw stones standing on the same position, the one throws farther win the game. Aha, of course, there are some naughty boys who care more about whether they can urinate father.

You believe it or not, anyway, I believed. Nowadays, some of the children are smarter than we were, while others may be more naughty.

A week ago, I saw several children throw stones. In fact, they are more clever than we were, since the game they played, apparently, is more complex than we did. Maybe you have different points of view, however, you'd better learn about the rules of the game before expressing your views. A group of children take turns to throw stones standing on the same position. After some child throw a stone, the children will draw a convex polyhedron with smallest volume together to enclose all the stones thrown by them. You may assume that the stone is so small as to be abstracted as a point in three-dimensional space. Naively, the children regard the space enclosed by the convex polyhedron as territory under their control. After a child throw his stone, the score he obtains equals the incremental of the volume of their territory.

Unfortunately, the first three child's score will always be zero. At last, the child with the highest score will win the game, and known as the "King".

I think you have accepted my opinion already, for the rules of their throwing stones game are really complicated. But, you also don't need to be frustrated for it. Now, in order to show you are smarter, maybe you can write a program to help the children point out their "King".

Input

Input consists of a number of cases. The data of each case appears on a number of input lines, the first of which contains an integer N. The following N lines contain three number (x_i, y_i, z_i) indicating coordinates of the stone thrown by the i-th child.
Note: 1 <= N<= 10^4, 1 <= i <= N, -10^4 <= x_i , y_i , z_i <= 10^4.

Output

For each test case, you should output two lines. The first line is "Case #K:", K means the number of the test case. The second line is "i v", i means index of the "King" and v means the score of the "King". If there are more than one "King", output the one throws stone earlier than others.

Please round the result to 2 digits after decimal point if necessary.

Sample Input

4
1 0 0
1 1 0
0 1 0
0 0 1
5
1 0 0
1 1 0
0 1 0
0 0 0
0 0 1

Sample Output

Case #1：
4 0.17
Case #2：
5 0.33

Problem J

Bombs

Description

Terrorists are around everywhere, they always make troubles by detonating bombs. The terrorist have some gunpowder to make bombs, different gunpowder has different damage, every kind of gunpowder can use any times, and the power of one bomb is the product of the gunpowder it consists of. Let's see how they make a bomb.

At the beginning they decide to use X parts of gunpowder to make a bomb, and then choose X parts of gunpowder, every time the damage of the gunpowder they choose can't be smaller than the last time they choose excepting the first time. After choosing X parts gunpowder terrorists get gunpowder[1], gunpowder[2] ... gunpowder[X] (gunpowder[1] <= gunpowder[2] <= ... <= gunpowder[X]), and then mix the X parts gunpowder to generate a bomb with power of the product of the damage of the gunpowder. Terrorists make bombs in some order, if they make bomb_A before bomb_B one of the following conditions should meet.

- Terrorists use less parts gunpowder to make bomb_A than bomb_B.
- Terrorists both use X parts of gunpowders to make bomb_A and bomb_B. There exist an integer j（j <=X）,for all i < j,gunpowder_A[i] = gunpowder_B[i] and gunpowder_A[j] < gunpowder_B[j].

Now, the police get the gunpowder by some way, police find that the gunpowder's damage is in the range of A to B(A, B included), police want to know the K-th bomb with the power in the range of L to R(L, R included).

Input

There are multiple cases, the first line is an integer *T* denoting the number of the case, for each case has five integers *A*, *B*, *L*, *R*, *K* in a line. *A*, *B* denote the damage range of the gunpowder. *L*, *R* denote the power range of the bomb, *K* denotes the *K*-th bomb with the power in the range *L* to *R* that police want to know.

2<=*A* <= *B*<=10^6

1<=*L*<=*R*<=10^9

1<=*K*<=10^6

Output

For each case output in the format in the first line "Case #x: y" x is the case number start from 1, y is the power of the bomb, and the second line with the gunpowder in the order they choose. If there is no more than K bombs in the range of L to R just output one line "Case #x: -1".

Sample Input

4
2 2 1 4 1
2 5 1 4 4
73 23642 12 20903 29401
2 50 1 1000000000 815180

Sample Output

Case #1: 2
2
Case #2: 4
2 2
Case #3: -1
Case #4: 59200
4 4 5 20 37

Hint

In the second case we have 4 kinds of gunpowder with damage 2, 3, 4, 5.
the first bomb is "2"with power of 2
The second bomb is "3" with power of 3
The third bomb is "4" with power of 4
The fouth bomb is "5" with power of 5
The fifth bomb is "2 2" with power of 2 * 2 = 4
So the 4-th bomb with power in the range of 1 to 4 is "2 2".

Problem K

Changsha Marathon

Description

Changsha is a beautiful city surrounded by mountains and rivers, forming a charming scenery. There are many tourist attractions in Changsha: Yuelu Academy, the predecessor of Hunan University, which is one of the four most famous academies in China. It was established in 976A.D. in Northern Song Dynasty. Juzizhou Island (Orange Isle) is the world's longest inland river isle which lies in Xiangjiang River and it is famous for the famous poet Qinyuanchun-Changsha written by Chairman Mao. Mawangdui is a famous archaeological site that contains the tombs of three people from the Western Han Dynasty. The tomb of Xin Zhui is best preserved among the three with a complete cosmetic set, lacquered pieces and finely woven silk garments. Huogongdian represents the Huxiang culture and Chuwu culture, Hunan's geography culture and Hunan's distinctive food culture. Tourists can have a taste of typical Hunan cuisine in it.

Every year the city holds a marathon. When planning the route, the organizers wish to go pass some tourist attractions to make it more enjoyable to attract more athletes and audience. Changsha can be treated as a graph consisted of N intersections and many bidirectional roads. A road connects two intersections i and j with length Wi,j. However, to reduce the impact to the daily traffic, only N-1 roads specified by the local government which can connect all N intersections are allowed to be used as the route. And the intersections with only one adjacent intersection are on the west bank of Xiangjiang River and there are ferries only at these intersections. The marathon starts at the intersection called Red East Square (marked as S) in Hunan University by convention. The organizers do not want the route pass any road twice, so the marathon has to be separated into two sections, the first section and the second section. Two different points have to be selected from intersections on the bank to act as the half end point S1 and the half starting point S2. The first section starts from S and end in S1. The second section starts from S2 and end in S. The organizer will use ferryboats to transport athletes from S1 to S2 as required by the government.

Considering the safety, some supply points are to be set on some intersections on each half section and there must be a supply point set on S, S1 and S2.

According to scientific research, the most suitable distance between two adjacent supply points is L and the organizers want to minimize the evaluation of the route. For each section, the evaluation from the starting point to rest point i is marked as Ei, and

$E_i = E_j + \sqrt{E_j} + (\sum_{k=j}^{i-1} D_{k,k+1} - L)^2$, where j is the previous supply point on the route, $\sum_{k=j}^{i-1} D_{k,k+1}$ means the sum of the roads' length on the route from intersection j to i. The evaluation of the starting point is zero. The evaluation of a section equals to the evaluation from its starting point to end point.

Your task is to select S1, S2, and the supply points to minimize the total evaluation of both sections.

Input

There are multiple test cases.

For each test case the first line contains two integers, N, L. ($1<=N<=20000$, $1<=L<=10^9$)

The next N-1 lines with 3 integers, U, V, W, describes that a bidirectional road connects intersection U and V with distance W. ($1<=U$, $V<=N$, $1<=W<=10^3$)
Red East Square (S) is always numbered 1. The input will finish with the end of file.

Output

For each case the output contains only one line, the evaluation of the best plan rounded to the second digit after the decimal point. If there is no solution, output -1.

Sample Input

```
7 4
1 2 2
1 3 7
1 4 3
2 5 6
2 6 5
3 7 8
```

Sample Output

```
7.00
```

Hint

The sample solution: