Problem A : ICPC

The Iranian ChamPions Cup (ICPC), the most prestigious football league in Iran, is reaching its end, and people are eagerly waiting for the finals, which happened to be between the two most popular Iranian teams, Persepolis and Esteghlal.

The ICPC finals consist of two matches, with each team competing as the home team in one match. The winning team is determined by aggregate score, the sum of the scores of the two matches. For example, if the scores of the two matches are Persepolis 6–0 Esteghlal in the first match, and Esteghlal 3–1 Persepolis in the second match, then the aggregate score will be Persepolis 7–3 Esteghlal, meaning that Persepolis is the winner. If aggregates are equal, the away goals rule is used to determine the winner, in which case the winner is the team that scored the most goals in the match it played away from home. If the result is still equal, a penalty shootout is required.

Hana, an avid football fan, is trying to figure out various scenarios in which her favorite team wins the finals. To this end, she aims to write a program that gets as input the number of goals in the two matches, and decides which team is the winner if it can be derived from the aggregate scores and the away goals rule, otherwise declares that the match goes to penalty kicks. You are going to help Hana write such a program.

Input

The first line of the input contains two space-separated integers p_1 and s_1, where p_1 and s_1 are the number of goals scored by Persepolis and Esteghlal, respectively, in the first match in which Persepolis is the home team. The second line contains two space-separated integers s_2 and p_2, where s_2 and p_2 are the number of goals scored by Esteghlal and Persepolis, respectively, in the second match in which Esteghlal is the home team. All input integers are between 0 and 20, inclusively.

Output

In the output, print the name of the winning team, either `Persepolis` or `Esteghlal`, if the winner can be determined by the aggregate scores and the away goals rule. Otherwise, print `Penalty`.

Example

Standard Input	Standard Output
`3 0` `2 1`	`Persepolis`

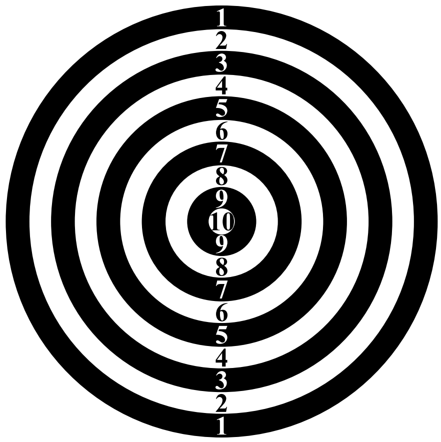
Standard Input	Standard Output
`3 1` `2 0`	`Esteghlal`

Standard Input	Standard Output
`2 0` `2 0`	`Penalty`

Problem B : Chaarshanbegaan at Cafebazaar

Chaarshanbegaan is a gathering event at Cafebazaar similar to TGIF events at Google. Some entertainment programs like pantomime, foosball, Xbox/PS4, and several board games are part of the event. You are going to set up a dart game in Chaarshanbegaan. As a techie organizing a game for techies, you would rather use a smart screen and write a program to calculate the scores instead of hanging a traditional dartboard and scoring the shots manually. Your program must get the coordinates of dart shots for a player and calculate his/her total score. The score for each dart shot (at point (x, y)) is calculated based on its distance from the center of the dartboard (point $(0, 0)$). If the distance is d millimeters, the score is calculated based on the following table:

Condition	**Score**
$d \leqslant 10$	10
$10 < d \leqslant 30$	9
$30 < d \leqslant 50$	8
$50 < d \leqslant 70$	7
$70 < d \leqslant 90$	6
$90 < d \leqslant 110$	5
$110 < d \leqslant 130$	4
$130 < d \leqslant 150$	3
$150 < d \leqslant 170$	2
$170 < d \leqslant 190$	1
$190 < d$	0



Input

The first line of the input contains a single integer N as the number of dart shots for a player ($1 \leqslant N \leqslant 100$). Each of the next N lines contains two space-separated integers as the coordinates (x, y) of a dart shot. The coordinates are in millimeters and their absolute values will not be greater than 300.

Output

Print a single line containing the total score of the player.

Example

Standard Input	Standard Output

`2`
`4 7`
`-31 -5`
	`18`

Standard Input	Standard Output

`3`
`12 -16`
`-180 100`
`152 10`
	`11`

Problem C : Cinema

The main movie theater of the city consists of a single auditorium with rows of comfortable padded seats. Surprisingly, the comfortness of seats are not necessarily equal. Precisely, each seat has its own comfort value which is a non-negative integer number. A seat is more comfortable than another seat if its comfort value is larger. It is only possible to enter a row from the left side of the auditorium. Assume the seats in a row are numbered 1 to m from left to right. When a person enters a row, he/she always sits on the most comfortable seat which is free and accessible to him/her. If he/she sits at seat i, he/she blocks other persons coming later to sit on seats $i+1$ to m. If there are more than one free and accessible seat being the most comfortable, he/she sits on the leftmost one. The owner of the movie theater plans to improve the comfortness of some seats to have more audiences in the auditorium. Improving one unit in the comfortness of a seat costs some fix value. With the budget available, the owner knows the total improvement over all seats must not exceed a value k. Help the owner find the best way to improve the comfortness of seats by at most k units in total to have the maximum number of audiences in the auditorium.

Input

The first line contains three non-negative integers n, m, and k ($1 \leqslant n \cdot m \leqslant 3 \times 10^5$, $0 \leqslant k \leqslant 10^{12}$) which are the number of rows, the number of seats in each row, and the total comfortness that can be added to all seats. The next n lines describe the comfort values of seats; each line contains m non-negative integers not more than 10^6 denoting the comfort values of seats from left to right for a row.

Output

Print a single line containing the maximum number of audiences.

Example

Standard Input	Standard Output
2 3 10 10 1 12 8 3 6	5

Standard Input	Standard Output
1 4 6 9 8 10 8	3

Standard Input	Standard Output
1 3 2 10 10 10	2

Problem D : Nader Shah

Nader Shah Afshar was one of the most powerful rulers in Iran. He won many battles, such as the battle of Herat, Mihmandust, and Kirkuk. During his ruling, Mashhad was Iran's capital. Each year, he planned an attack on a new country, and annexed it to Iran's territory. As Nader Shah's past victories were known to everyone, the country under attack was surrendering the battle without any combat. Therefore, after some years, Iran's territory was expanded to n new countries.

There was a connected road network between the countries. All roads were two-way passing. Nader Shah only attacked those countries which had at least one road to Iran's territory at the moment. After capturing each country, Nader Shah chose exactly one road between the captured country and Iran's territory at that moment, and put Iran flag on it at every one kilometer. In order to make costs as minimal as possible, he chose the shortest road as it needed less Iran flags (if there are more than one shortest road, he arbitrarily chose one of them). These roads were called Afshari roads. Obviously, by choosing the roads this way, there are exactly $k-1$ Afshari roads when Iran's territory consists of k countries.

Several years later, we have received a map of Iran's territory at the end of Nader Shah's life. The map has all the road network on it with their lengths in kilometers. Afshari roads are marked in the map. There are n countries on the map, m roads connecting them, and $n-1$ of those roads marked as Afshari. Each country is marked by a distinct number $1 \leqslant x \leqslant n$. We do not distinguish between the numbers and the countries.

Given the map described above, we want to know the possible country corresponding to Iran and the order by which Nader Shah captured countries. One possible answer can be shown as $a_1, a_2, \ldots, a_n$ where a_1 corresponds to Iran, a_2 is the first country that Nader Shah captured, a_3 is the second country he captured, and so on. We want to know the lexicographically minimum possible answer. Array $x_1, x_2, \ldots, x_n$ is lexicographically smaller than $y_1, y_2, \ldots, y_n$ if and only if there exists a number i $(1 \leqslant i \leqslant n)$ that $(x_1 = y_1) \wedge (x_2 = y_2) \wedge \ldots \wedge (x_{i-1} = y_{i-1})$ and $x_i < y_i$.

Input

The first line of the input contains two integers n and m $(1 \leqslant n \leqslant 1,000,\ n-1 \leqslant m \leqslant 5,000)$ where n and m are the number of countries and the number of roads in the map (the road network), respectively. The next m lines describe the road network; each line contains four integers u, v, w, and r $(1 \leqslant u, v \leqslant n,\ 1 \leqslant w \leqslant 10^6$, and $r \in \{0, 1\})$ which indicate there is a road between u and v with the length of w kilometers, and r denotes whether the road is Afshari or not. The road is Afshari if and only if $r = 1$. There is at most one road between two countries. It is guaranteed that there are exactly $n-1$ Afshari roads in the road network, and all n countries are connected through these $n-1$ roads.

Output

If there is no answer to the problem, print `Wrong Map!`. Otherwise, write the lexicographically minimum answer to the problem.

Example

Standard Input	Standard Output
3 3 1 2 2 1 2 3 2 1 1 3 1 0	Wrong Map!

Standard Input	Standard Output
3 3 1 2 1 1 2 3 2 0 1 3 3 1	1 3 2

Problem E : JackRabbit Slim

We all love rabbits, right? Unfortunately, they don't even like us, rather, they love carrots instead!

They love carrots so much that they would do anything to get to a carrot, even running. That might sound easy, but rabbits are lazy, it's the jackrabbits who are the active ones. That's why this problem is about jackrabbits and not rabbits. In fact, it's about a very specific black-tailed jackrabbit, named Slim.

Earlier this morning, before the contest started, a truck loaded with carrots passed through the road by Slim's house. Slim was so lucky that n carrots dropped off on the road. We number them 1 to n from left to right. When Slim left home, he found a carrot. After taking a look around, his plan for the day became clear: Locate the closest carrot, get to it and eat it, then repeat.

The road by Slim's house is a straight road of length 10^9 and the i-th carrot on the road is at coordinate x_i.

Consider Slim starts his day by standing at the position of the j-th carrot and eating it. Then as long as there are more carrots, he follows the following steps:

- Locate the closest remaining carrot. If there are two closest carrots, choose the one to the right.
- Run to it and eat it. Simple!

For each value of j, we call D_j the total distance Slim runs if it starts the day by eating carrot j. For an unknown reason, we are interested in finding the sum of all D_j values, for all values of j $(1 \leqslant j \leqslant n)$.

Input

The first line of the input contains a single integer n ($1 \leqslant n \leqslant 10^5$), the number of carrots. The second line of the input contains n space-separated distinct integers $x_1, \ldots, x_n$ ($0 \leqslant x_i \leqslant 10^9$). The coordinates of the carrots are in an increasing order.

Output

Output an integer value, which is the sum of all D_j values (for j from 1 through n).

Example

Standard Input	Standard Output
6 1 14 19 20 21 24	168

Problem F : Imputation

Leila is a Bioinformatician, interested in studying Bacterial evolution. In one experiment on a special type of Bacteria, she started from a single bacterium, put it on a plate, and monitored the bacterial division, until she obtained a population of k bacteria. During the process, she carefully reported the evolutionary relations between bacteria. Precisely, for each bacterium, she reported its parent bacterium.

In the next step, she extracted DNA sequences of k bacteria in the final population, by NGS technology. Each DNA sequence is represented as a string of length m from the alphabet set $\{A, T, C, G\}$.

The NGS technology has a drawback: it produces a lot of missing values. So, there are a lot of unknown characters indicated by '?' in the extracted sequences. Considering the evolutionary relationship between bacteria, Leila wants to impute the missing values. Among all possible imputations, she wants to find the minimum cost imputation from an evolutionary perspective.

The problem is defined as follows. A rooted tree T is given, and for each leaf v of T, a string b_v of length m from the character set $\{A, T, C, G, ?\}$ is given. A transition cost matrix Δ is also given, where $\Delta(x, y)$ $(x, y \in \{A, T, C, G\})$ represents the cost of a transition from an x character to a y character, from a parent to its child.

A feasible imputation, assigns a string s_u of length m from the character set $\{A, T, C, G\}$ to each vertex u, where for each leaf v of T, s_v is equal to b_v except for '?' characters in b_v. The evolutionary cost of an imputation is defined as the sum of evolutionary costs of all edges. The evolutionary cost of an edge between parent u and child w, is defined as $\sum_{i=1}^{m} \Delta(s_u[i], s_w[i])$, where $s_u[i]$ is the i-th character of s_u.

Leila wants to find a feasible imputation for T, which has the minimum evolutionary cost among all feasible imputations. The tree T, transition cost matrix Δ, and a string b_v for each leaf v are given. You should write a program to compute the minimum evolutionary cost of feasible imputations.

Input

The first line of the input contains an integer n $(2 \leqslant n \leqslant 10{,}000)$ denoting the number of vertices of T. The vertices of T are numbered from 1 to n. The root of the tree is numbered 1. The root is never considered as a leaf, even if it has only one child. The next $n - 1$ lines describe the edges of T; each line contains two endpoints of an edge separated by spaces. In the next four lines, the evolutionary cost matrix Δ is given; each line is for one row of Δ. Rows (corresponding to a parent) and columns (corresponding to a child) of Δ are ordered to respectively represent characters A, T, C and G. All entries of Δ are non-negative integers not more than 10^6. The next line just contains k, the number of leaves. Finally, each leaf v (its number) and its b_v which is a string of size m $(1 \leqslant m \leqslant 200)$ appear in one line.

Output

In one line, print the minimum evolutionary cost of feasible imputations.

Example

Standard Input	Standard Output
3 1 2 1 3 0 3 4 4 4 0 4 4 4 4 2 4 1 1 1 0 2 2 AAC 3 T?C	4

Problem G : Relay Race

In a relay running race, n athletes of a team are initially positioned along a road. Specifically, the initial position of athlete i is x_i (in meters from an origin). This athlete can run up to v_i meters per second. Initially, each athlete holds a baton.

The race starts by blowing a whistle and finishes when any athlete of the team holds all of the batons. In any moment during the race, each athlete can run along the road (in any of the two directions), or simply stop. When two athletes meet at the same position, each of them can pass all batons she or he holds to the other athlete.

You are the coach of the team. Your task is to find the shortest possible time the team can finish the race.

Input

In the first line of input, a single integer n ($1 \leqslant n \leqslant 10^5$) is given. In each of the next n lines, two space-separated integers x_i ($0 \leqslant x_i \leqslant 10^6$), and v_i ($1 \leqslant v_i \leqslant 10^6$) are given.

Output

In the only line of the output, print a single number, the minimum amount of time the team can finish the race (in seconds). Your answer is considered to be correct if it has an absolute error of at most 10^{-6}.

Example

Standard Input	Standard Output
3 4 1 0 3 10 1	2.50000

Standard Input	Standard Output
1 100000 1000000	0

Problem H : T-net

T-net which is a new telecommunications company, plans to install its base stations in the city. The places where base stations must be installed have been already specified. T-net has two types of antennas to be used in the base stations: (i) antennas with transmission radius a, and (ii) antennas with transmission radius b. Two antennas can communicate with each other if and only if both are inside the coverage area of each other. Antenna with smaller transmission radius of course is cheaper. T-net plans to minimize its cost while keeping the whole network connected. Precisely, T-net plans to minimize its cost which is the sum of the transmission radii of all antennas. Interestingly, all base-station places are on a line. Help T-net construct a connected network with the minimum cost.

Input

The first line of the input contains three positive integers n, a and b ($1 \leqslant n \leqslant 10^5$ and $1 \leqslant a, b \leqslant 10^5$) where n is the number of base stations, and a and b are radii as defined in the problem statement. The second line contains n distinct coordinates of base stations on the line with respect to an origin on the line. All coordinates are positive integers not more than 10^5.

Output

If it is possible to construct a connected network, print the minimum cost in the output. Otherwise, print `-1` .

Example

Standard Input	Standard Output
3 1 3 1 4 3	7

Problem I : Congestion Charging Zone

Tehran municipality has set up a new charging method for the Congestion Charging Zone (CCZ) which controls the passage of vehicles in Tehran's high-congestion areas in the congestion period (CP) from 6:30 to 19:00. There are plate detection cameras inside or at the entrances of the CCZ recording vehicles seen at the CCZ. The table below summarizes the new charging method.

The first time seen in the CP	The last time seen in the CP	Charge
6:30 to 10:00	6:30 to 16:00	24000
6:30 to 10:00	16:01 to 19:00	36000
10:01 to 16:00	10:01 to 16:00	16800
10:01 to 19:00	16:01 to 19:00	24000

Note that the first time and the last time that a vehicle is seen in the CP may be the same. Write a program to compute the amount of charge of a given vehicle in a specific day.

Input

The first line of the input contains a positive integer n ($1 \leqslant n \leqslant 100$) where n is the number of records for a vehicle. Each of the next n lines contains a time at which the vehicle is seen. Each time is of form <hour>:<minute>, where <hour> is an integer number between 0 and 23 (inclusive) and <minute> is formatted as an exactly two-digit number between 00 and 59 (inclusive).

Output

Print the charge to be paid by the owner of the vehicle in the output.

Example

Standard Input	Standard Output
4 7:30 2:20 7:30 17:30	36000

Standard Input	Standard Output
1 12:13	16800

Standard Input	Standard Output
2 0:30 23:30	0

Problem J : Image Processing Project

You want to be recruited in Cafebazaar, a leading IT company in Iran, as part of an image processing project for detecting celestial bodies. To prove yourself, you should do a pilot project in a matter of hours for the project manager.

The project manager has put some paper sheets on a rectangular table whose color is white. All paper sheets are square and have the same size. Moreover, all paper sheets are white but their boundaries are black. Each side of all paper sheets is parallel to an edge of the table. You must write an image processing program to count the number of paper sheets by receiving a picture of the table taken from above.

Input

The first line of the input contains two positive integers r and c ($3 \leqslant r, c \leqslant 200$) denoting the number of rows and columns of the given picture, respectively. Precisely, the picture is an $r \times c$ table of pixels. The next r lines, each contains exactly c characters. Each character is `+`, `-`, or `.` representing a pixel of the picture. A dash pixel illustrates one unit of a horizontal side of a paper sheets. A plus pixel illustrates either one unit of a vertical side of a paper sheet or a corner of a paper sheet. A dot pixel is a white pixel.

You can assume that in each row there exists at most one horizontal side. Similarly, in each column, there exists at most one vertical side. It is guaranteed that at least two corners of each paper sheet are visible. A corner is visible when along with the pixel corresponding to the corner, at least one immediate pixel from each of its incident edges is also present in the input. Moreover, It is guaranteed that there is at least a dash pixel in the input.

Output

Print the number of paper sheets in the output.

Example

Standard Input	Standard Output

`10 13`
`+---+........`
`+...+........`
`+...+...+---+`
`+...+-+.+...+`
`+---+.+.+...+`
`..+..+--+...+`
`..+..+..+---+`
`..+--+...+...`
`.....+...+...`
`.....+---+...`
	`4`

Problem K : Python Classes

Mr. Programmer has been busy working on his awesome startup project lately. The project was coded in Python. In the past few days, he was hugely refactoring the source code without any testing. After all changes have been made, he tried to run the project, but he figured out that the code is not executing properly anymore due to some cyclic dependencies created in his source code. He struggled a little bit moving things around only to know he is just messing up things even further. Maybe if I move all the classes to a single file I'll be able to resolve the problem, he thought. But that plan didn't work either. With a great frustration and feeling pressure to launch his project as soon as possible, he has finally decided to outsource the task of putting things in the correct order to algorithmic specialists.

The source code is a single Python file. It consists of a bunch of classes. Mr. Programmer doesn't like multiple inheritances much, so each class in his code has at most one super class. As he does not want to leak the content of the classes, he has nulled out the contents and effectively replaced them with the "pass" command in Python. Therefore, when a class does not have a superclass, it looks like below:

```
class <CLASS_NAME>:
    pass
```

When a class has a superclass, it looks like below:

```
class <CLASS_NAME>(<SUPER_CLASS_NAME>):
    pass
```

In the above codes, *CLASS_NAME* and *SUPER_CLASS_NAME* are arbitrary Python identifiers. For further clarification look at the sample input.

We know that the file is executable if for each class having a super class, its superclass appears somewhere before the class in the file. Mr. Programmer doesn't like too many changes to his file, so he wants to put the file in the correct order with the minimum number of changes. Each change is counted as cutting a single class (along with its content) and pasting it in some other position in the file. Help Mr. Programmer find the minimum number of changes to file so that the file becomes executable.

Input

The input is a python file. It consists of some python classes separated by a single blank line. The classes in the input are exactly in the format described in the problem statement. Class identifiers only consists of English upper or lower case letters, and their length does not exceed 10 characters (for the sake of this problem you should not make any other assumptions about the identifiers). There are exactly 4 space characters before each "pass" command. There are at most $2,000$ classes in the file. It is guaranteed that all super classes are defined somewhere in the file, and no class is defined multiple times. Furthermore, a class is not its own superclass.

Output

Print a single integer as the minimum number of changes that have to be made so that the file becomes executable, or `-1` if it is not possible to do so.

Example

Standard Input	Standard Output

class B(A):
    pass
 	
class C(A):
    pass
 	
class A:
    pass
	1