

Problem A. Campaign

Input file: `stdin`
Output file: `stdout`
Time limit: 2 seconds
Memory limit: 256 mebibytes

Donald Campaign has n members which are labeled from 0 to $n - 1$. Members use encrypted protocols to communicate with each other. There are m encrypted communication protocols. The i -th one is an encrypted protocol used by u_i and v_i to communicate directly with each other.

For the i -th protocol, Bill knows u_i and v_i . However he doesn't know the password.

Recently, he has penetrated into campaign server. He found out that there is a file in campaign server that contains password of each protocol in separate lines. The i -th line, contains decryption password of the i -th protocol. However he can only read some consecutive lines of the file before being disconnected by server's firewall. He will consider his attempt *successful*, if using discovered passwords, he can send message from every member to every other member using one or more communication protocols. (member A can send message to member B , if there is a direct communication protocol between them, or there exists some member C which A can send message to C and C can send message to B)

Bill is too cautious, he wants to know number of possible *successful* attempts before his attempt.

Input

The first line contains two integers n and m ($1 \leq m \leq 10^5$) – number of members and number of communication protocols. The i -th of next m lines contains u_i and v_i ($0 \leq u_i, v_i \leq n - 1$, $u_i \neq v_i$), which describes i -th communication protocol.

Output

Print number of possible *successful* attempts.

Scoring

Subtask 1 (36 points)

$n = 4$

Subtask 2 (64 points)

$1 \leq n \leq 10^5$

Examples

stdin	stdout
3 5 0 1 1 2 2 1 0 1 0 2	9

Problem B. The Pursuit of Happiness

Input file: `stdin`
Output file: `stdout`
Time limit: 1 seconds
Memory limit: 256 mebibytes

Jintan and Menma live in Treeland. Treeland is a country consisting of n cities. There are $n - 1$ bidirectional roads connecting the cities and there exists a path between every pair of cities.

Menma knows that Jintan loves playing video games. She recently got word that there are m video games sold around the country. Each game has two installation discs. She can install a game if she has both of the discs. Menma knows she can buy the first and the second disc of the i -th video game at cities u_i and v_i respectively. She also knows that installing the i -th video game will increase Jintan's happiness by w_i units.

Menma is going to choose two cities A and B, and she's going to buy every disc that is being sold on the simple path between A and B for Jintan. She wants to increase Jintan's happiness as much as possible. Help her find the maximum value of Jintan's happiness if she chooses A and B optimally. (Jintan's happiness is currently equal to 0)

Input

The first line contains two integers n and m ($1 \leq n \leq 10^5$, $1 \leq m \leq 10^5$) – the number of cities and the number of video games, respectively.

The next $n - 1$ lines contain description of the roads. The i -th ($0 \leq i \leq n - 2$) line of these lines contains a single integer p_i ($0 \leq p_i \leq n - 1$), which means there is a bidirectional road between cities p_i and $i + 1$.

The next m lines describe the video games. The i -th ($0 \leq i \leq m - 1$) of these lines will contain three integers u_i , v_i and w_i respectively. ($0 \leq u_i, v_i \leq n - 1$, $u_i \neq v_i$, $0 \leq w_i \leq 10^4$)

Output

Print the maximum value of Jintan's happiness if Menma chooses A and B optimally.

Examples

stdin	stdout
4 3 0 0 0 0 2 2 1 3 99 1 2 98	100

Problem C. Smurfs

Input file: `stdin`
Output file: `stdout`
Time limit: 2 seconds
Memory limit: 256 mebibytes

What a great life! n Smurfs live on a big and green tree. They are labeled from 0 to $n - 1$ and each live in a different house (on the tree). Gargamel, the evil wizard, always tries to offend them. Now with Hogata and Baltazar he wants to destroy their big tree. Their tree has $n - 1$ branch which the i -th one connects u_i -th Smurf house to v_i -th. There exists a path between every pair of Smurfs houses.

Only *Great Smurf* can defeat Gargamel. Therefore, if a Smurf house can't reach any *Great Smurf* house, it'll be destroyed. (Hence a *Great Smurf* never dies).

Knowing this fact, Gargamel decided to cut some of branches and destroy Smurfs. Hogata and Baltazar, do the cutting as a game.

In the game, Hogata and Baltazar cut branches in turn. After each cut, each house that can't reach any *Great Smurf* house will be destroyed along with branches connected to it. The one who can't make a move in his turn, loses the game. Baltazar starts first.

Gargamel asked you to write a program that computes who will win the game if they both play optimally.

Input

The first line contains integer t ($1 \leq t \leq 5$) – number of testcases.

For each testcase, first line contains n ($1 \leq n \leq 2 \times 10^5$) – number of Smurfs in tree. The following line contains n binary numbers a_0, a_1, \dots, a_n that a_i is 1 if i -th Smurf is a *Great Smurf* and 0 otherwise. i -th of the following $n - 1$ lines contains u_i and v_i ($0 \leq u_i, v_i \leq n - 1$) which means there is a branch between u_i -th house and v_i -th house.

Output

For each testcase on a separate line, print "Baltazar" if Baltazar wins and "Hogata" otherwise. (without quotation marks).

Scoring

Subtask 1 (24 points)

There is only one *Great Smurf* in each testcase.

Subtask 2 (76 points)

No additional limits.

Examples

stdin	stdout
2	Baltazar
4	Hogata
1 0 0 1	
0 1	
1 2	
2 3	
5	
1 0 0 0 0	
0 1	
1 2	
1 3	
0 4	

Problem D. Lazy Reimu

Input file: `stdin`
 Output file: `stdout`
 Time limit: 5 seconds
 Memory limit: 256 mebibytes

Reimu was at a shrine; when she opened the door she stumbled upon a box. Thinking it is related to the recent hot weather, she tried to open it but it was locked. Then she noticed some writings on the box:

If you wish to know the password of the lock you have to solve the following problem:

You have a circle of n binary numbers, ($0 \leq a_i \leq 1$ for $0 \leq i \leq n - 1$) You also have another array c of n distinct numbers with size k , ($0 \leq c_i < n$ for $0 \leq i \leq k - 1$). There is a button, if you press it, the numbers will change. If before pressing button i -th number was a_i it will change into b_i :

$$b_i = a_{(i+c_0) \bmod n} \oplus a_{(i+c_1) \bmod n} \oplus \dots \oplus a_{(i+c_{k-1}) \bmod n}$$

You have to press the button t times, and the final numbers will be the password for the box.

Reimu is too lazy to solve the problem with her magic, so she gave it on TST contest. Now you have to solve the problem for her.

Input

The first line of input contains a string with n ($1 \leq n \leq 10^5$) character. a_i is 1 if i -th letter is '1' and 0 otherwise.

The second line of input contains another string with n character with exactly k ($1 \leq k \leq n \leq 10^5$) '1' letter. if the i -th letter is '1', then array c contains i .

The third and last line of input, contains string ts ($1 \leq |ts| \leq 10^5$) which is number t in the binary representation. ts will contain at most 15 one.

Output

Print a string with length n which i -th letter shows the i -th number after pressing the button t times.

Scoring

Subtask 1 (75 points)

$$1 \leq k \times n \leq 10^5$$

Subtask 2 (25 points)

No additional limits.

Examples

stdin	stdout
1010 0100 1	0101

Problem E. Padeshash

Input file: `stdin`
Output file: `stdout`
Time limit: 2 seconds
Memory limit: 256 mebibytes

A film festival is taking place and Ali is appointed to be the judge. He has decided to watch the movies a_0, a_1, \dots, a_{n-1} , in the given order. Note that a movie may appear more than once in the sequence.

There are m theaters; the i -th is showing the set V_i of the movies. Each day Ali should choose a theater x and pay p_x dollars to enter the theater. Then he can watch any sequence of any length consisting of the movies in V_x . Also it is possible that he watches a movie multiple times in a theater.

Ali wants to know the minimum amount of money he needs to watch the movies on his list, in the given order. Note that the festival is going to last for a very long time, so there is no limit on the number of days Ali takes to watch the movies.

Input

The first line contains two integers n and m , the length of Ali's sequence and the number of the theaters.

The second line contains n integers a_0, a_1, \dots, a_{n-1}

Then in the $(2+i)$ -th line are the numbers p_i and $size_i$, followed by $size_i$ distinct integers $V_{i,j}$ (the movies on the i -th theater.)

Output

In the only line of output print one integer, the minimum amount of money needed to watch the movies.

Constraints

$$1 \leq n, m \leq 10^5$$

$$0 \leq \sum_{i=1}^n size_i \leq 10^5$$

$$0 \leq a_i, V_{i,j} \leq 10^5$$

Subtask 1 (30 points)

$$0 \leq a_i, V_{i,j} < 20$$

Subtask 2 (70 points)

No additional limits.

Examples

stdin	stdout
8 4 1 4 3 1 2 4 3 3 10 2 1 2 10 3 1 4 3 20 2 4 3 1 5 1 2 3 4 5	40

Problem F. Aseman

Input file: `stdin`
Output file: `stdout`
Time limit: 30 seconds
Memory limit: 256 mebibytes

Corruption of an organization has been revealed! People have gathered in front of the headquarters to protest against them. n rebels are going to throw stones at the building! The building has k windows, h of which are double-paned (i.e. with the width twice the width of a single-paned window) and the rest are single-paned. The heads of the company have hired m people and gave each of them a piece of glass to make one of the windows stronger. Note that strength of a window can get arbitrarily large and become x -paned. Each of m people, randomly and independently of others' choices or numbers of window panes, chooses a window and makes it stronger.

Then each of the rebels, randomly and independently of others' choices or numbers of window panes, chooses a window and throws a stone at it. Hence one of its glasses breaks. If all glasses of at least one window breaks, the building will become unsafe. Write a program to calculate the probability that the building will be safe.

Input

The first line contains t – number of testcases.

Each testcase consists of one line containing k, n, m, h – number of windows, number of rebels, number of people hired to make windows stronger and number of double-paned windows respectively.

Output

For each testcase print one line containing the probability that building will be safe with absolute error less than 10^{-6} .

Constraints

$$1 \leq t \leq 200$$

$$1 \leq k \leq 200$$

$$0 \leq h \leq k$$

$$1 \leq n, m \leq 100$$

Subtask 1 (30 points)

$$k \leq 20$$

$$n, m \leq 30$$

$$t \leq 100$$

Subtask 2 (70 points)

No additional limits.

Examples

stdin	stdout
3	0.333333333
3 1 1 0	0.000000000
3 2 1 0	0.851851852
3 1 2 2	

Problem G. Damdaran

Input file: `stdin`
Output file: `stdout`
Time limit: 1 seconds
Memory limit: 256 mebibytes

Recently a new kind of cow has been discovered which is called “Shy cow”; it’s because these cows only eat when other cows can’t see them. To raise these cows, a special pasture with n green points has been made. The i -th ($0 \leq i \leq n - 1$) green point is located at (i, y_i) .

Two cows can see each other if none of the green points between them, is **strictly** above the line connecting them. For example in the second sample case the first and the last cow can see each other. It is obvious that two cows can see each other at the same point.

We need to place maximum number of cows at the green points so that no two can see each other. Write a program to calculate this number.

Input

First line of input contains n – number of cows.

The second line contains n integers y_0, y_1, \dots, y_{n-1} .

Output

Print maximum number of cows that can be placed that no two can see each other.

Constraints

$$1 \leq n \leq 2000$$

$$0 \leq y_i \leq 10^9$$

Scoring

Subtask 1 (70 points)

$$n \leq 200$$

Subtask 2 (30 points)

No additional limits.

Examples

<code>stdin</code>	<code>stdout</code>
6 0 10 0 5 6 11	333
3 0 1 2	1

Problem H. Noble Prize

Input file: `stdin`
Output file: `stdout`
Time limit: 4 seconds
Memory limit: 512 mebibytes

Ilich is about to win the Noble Prize in Gardening. The noble prize awards show is being held in a garden in Oslo, Norway. The host of the noble prize awards still doubts Ilich's gardening skills and is going to test him with a series of questions about the garden they're in. This garden has n trees numbered from through 1 to n , each one having coordinates (x_i, y_i) . No two trees share the same coordinates, and no three of them are co-linear (lie on a line). The noble prize host will ask Ilich questions. In each question he gives him the index of three different trees in the garden and Ilich must tell him the number of trees that lie inside the triangle whose vertices are the three given trees (the vertices themselves don't count).

Ilich is a simple gardener and has no knowledge of geometry whatsoever, so he asked for your help. Help Ilich get his noble prize.

Input

The first line contains an integer n , the number of trees in the garden ($3 \leq n \leq 2000$).

The next lines contain the coordinates of the trees, each one contains two integers x_i and y_i , separated by space ($-10^9 \leq x_i, y_i \leq 10^9$).

The next line contains an integer q , the number of questions ($1 \leq q \leq 5 \times 10^5$).

The next q lines contain the coordinates, each one contains three integers a_i, b_i , and c_i , the index of three of three trees in the garden, separated by space ($1 \leq a_i, b_i, c_i \leq n, a_i \neq b_i, a_i \neq c_i, b_i \neq c_i$).

Output

For each question, print the number of trees that lie inside the triangle made by the trees in the question, in one line.

Examples

stdin	stdout
3 0 0 0 1 1 0 1 1 2 3	0
4 -1 -1 1 -1 0 1 0 0 4 1 2 3 1 2 4 1 3 4 2 3 4	1 0 0 0

Problem I. Killshot

Input file: `stdin`
Output file: `stdout`
Time limit: 4 seconds
Memory limit: 512 mebibytes

"Charpa just hit me on a text, last night I left a problem on her desk. said Reza. He's decided to punish Charpa for cheating in the online contest of ChefCPC. As expected, Reza gave their team this problem, and their team will be disqualified from ChefCPC for life if they don't solve this! Here's the problem:

Given an integer k , determine if there is a simple, connected, bidirectional graph G such that the sum of distances between all pairs of vertices in G equals k . Distance between two vertices x and y is the number of edges on the shortest path from x to y .

If such a graph exists, print it in the output.

Input

The first and only line of output contains a single integer k ($1 \leq k \leq 10^5$)

Output

If there's no such graph, print -1 .

Otherwise: The first line of output should contain a single integer n .

n should be $n(G)$. The second line should contain a single integer m , the number of edges in G .

The next m lines should contain the edges, each line should contain two integers v and u ($1 \leq v, u \leq n$ and $v \neq u$). There should be no more than one edge between any pairs of vertices.

If there are multiple solutions, print any of them.

Examples

stdin	stdout
4	3 2 1 3 2 3
12	5 8 2 5 2 4 4 5 3 5 1 4 3 4 2 3 1 5
2	-1

Problem J. Into the Baladlands

Input file: `stdin`
Output file: `stdout`
Time limit: 4 seconds
Memory limit: 512 mebibytes

Bardia just got hired by Balad as their "Navigation Expert". His job is simply to navigate! Every time a user wants to navigate to a destination, their client app sends a request to Bardia, then he finds the optimal path to that destination and sends it back to user's app. Taki's really mad at Bardia for prank calling him, so he decided to take revenge by sending him really hard navigation requests. He knows that Bardia's familiar with all major cities and can easily navigate in them. So he chose a city that Bardia hasn't seen before, the Baladland.

Baladland has only 4 streets, which are parallel and vertical, named A, B, C and D. A is the $x = 1$ line, B is $x = 2$, C is $x = 3$ and D is $x = 4$. All these streets are one-way; more precisely, A and C are south-to-north (meaning you can only go upward) and B and D are north-to-south.

There are n u-turns in Baladland. Using a u-turn you can only go to the next street on your right, meaning that using a u-turn on A street you can only go to B street, using a u-turn on B street you can only go to C street, and using a u-turn on C street you can only go to D street. There are no u-turns on D street. Each of these u-turns has coordinates (x_i, y_i) , meaning it's located on $y = y_i$ on the street with $x = x_i$ and ends up on $y = y_i$ on the next street (with $x = x_i + 1$). For example a u-turn with $(x_i, y_i) = (2, 10)$, is On B Street at $y = 10$ and using it you will end up on $(3, y_i)$ which is on C street at $y = 10$.

Going one unit (one meter) up or down on a street will take 1 second. Using a u-turn also takes 1 second. Taki is located on $(1, s)$ (on A street) and his destination is $(4, t)$ (on D street). Help Bardia tell him how many seconds it takes for Taki to reach his destination, if possible.

Input

The first line of input contains three integers n , and s ($1 \leq n \leq 10^5$, $1 \leq s, t \leq 10^9$).

The next n lines contain the u-turns. Each one contains two integers x_i and y_i ($1 \leq x_i \leq 3$, $1 \leq y_i \leq 10^9$).

It's guaranteed that the numbers $s, t, y_1, y_2, \dots, y_n$ are all distinct.

Output

Print the minimum time it takes for Taki to reach his destination, or -1 if it's impossible.

Examples

stdin	stdout
5 4 7 1 5 1 8 2 6 2 1 3 9	14