Kotlin Heroes: Episode 2

A. Three Problems

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Polycarp is choosing three problems for creating a programming test. Totally he has n problems in his list. The complexity of the i-th problem equals r_i. All problems are numerated from 1 to n.

Help Polycarp to choose such three problems a, b and c, so that the complexity of the first problem strictly less than the complexity of second problem and the complexity of the second problem is strictly less than the complexity of the third problem. So, for chosen problems a, b and c it should be true that $r_a < r_b < r_c$.

If Polycarp can choose three problems in different ways, you can print any of them.

Input

The first line of the input contains one integer n ($3 \le n \le 3000$) — the number of problems in Polycarp's list.

The second line of the input contains n integers $r_1, r_2, \dots, r_n$ ($1 \le r_i \le 10^9$), where r_i is the complexity of the i-th problem.

Output

If Polycarp has no ways to choose three problems, you should print three numbers -1. Ih there is a way to choose them, you should print three different integers a, b, c ($1 \le a, b, c \le n$), where a is the number of the first chosen problem, b is the number of the second chosen problem and c is the number of the third chosen problem.

Examples

input

```
6
3 1 4 1 5 9
```

output

```
4 1 3
```

input

```
5
1 1000000000 1 1000000000 1
```

output

```
-1 -1 -1
```

input

```
9
10 10 11 10 10 10 10 10 1
```

output

```
9 8 3
```

B. Traveling Around the Golden Ring of Berland

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

The Golden Ring is the special tourist route in Berland. This route consists of n cities and the cyclic railway route. Cities are numbered from 1 to n so that:

- the next city for 1 is the city 2,
- the next city for 2 is the city 3,
- ...
- the next city for n is the city 1.

Thus, the route is a cycle, cities are numbered in the direction of travel (the route is directed in one way).

Blogger Polycarp wants to start his journey in the city 1. For each city he knows the value a_i — how many selfies he wants to do in i-th city. He can take no more than one selfie in one visit to each city. Since he is traveling by train, he can't skip the city (he always moves from the city i to the city $i+1$ for $1 \le i < n$ and from n to 1). Thus, when the train stops in the city, Polycarp must visit this city. If Polycarp visits the city multiple times, all visits are counted separately.

What is the least number of city visits Polycarp will have to complete to fulfill his plan for the number of selfies for each city? Note that he always visits the city 1, since it is this city that his journey begins in.

Input

The first line contains an integer n ($3 \le n \le 2 \cdot 10^5$) — the number of cities in the Golden Ring of Berland.

The next line contains n integers $a_1, a_2, \dots, a_n$ ($0 \le a_i \le 10^9$), where a_i is equal to the required number of selfies in the i-th city.

It is guaranteed that at least one of the numbers a_i is strictly greater than zero.

Output

Print a single integer — the required minimum number of visits.

Examples

input
```
3
1 0 0
```
output
```
1
```

input
```
3
2 0 2
```
output
```
6
```

input
```
5
0 3 1 3 2
```
output
```
14
```

input
```
5
1000000000 1000000000 1000000000 1000000000 0
```
output
```
4999999999
```

C. Ice Cream

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Summer in Berland lasts n days, the price of one portion of ice cream on the i-th day is c_i. Over the summer, Tanya wants to eat exactly k portions of ice cream. At the same time, on the i-th day, she decided that she would eat at least a_i portions, but not more than b_i ($a_i \le b_i$) portions. In other words, let d_i be equal to the number of portions that she eats on the i-th day. Then $d_1 + d_2 + \dots + d_n = k$ and $a_i \le d_i \le b_i$ for each i.

Given that portions of ice cream can only be eaten on the day of purchase, find the minimum amount of money that Tanya can spend on ice cream in the summer.

Input

The first line contains two integers n and k ($1 \le n \le 2 \cdot 10^5$, $0 \le k \le 10^9$) — the number of days and the total number of servings of ice cream that Tanya will eat in the summer.

The following n lines contain descriptions of the days, one description per line. Each description consists of three integers a_i, b_i, c_i ($0 \le a_i \le b_i \le 10^9$, $1 \le c_i \le 10^6$).

Output

Print the minimum amount of money that Tanya can spend on ice cream in the summer. If there is no way for Tanya to buy and satisfy all the requirements, then print -1.

Examples

input
```
3 7
3 5 6
```

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

output

```
31
```

input

```
1 45000
40000 50000 100000
```

output

```
4500000000
```

input

```
3 100
2 10 50
50 60 16
20 21 25
```

output

```
-1
```

input

```
4 12
2 5 1
1 2 2
2 3 7
3 10 4
```

output

```
35
```

Note

In the first example, Tanya needs to eat 3 portions of ice cream on the first day, 1 portions of ice cream on the second day and 3 portions of ice cream on the third day. In this case, the amount of money spent is $3 \cdot 6 + 1 \cdot 4 + 3 \cdot 3 = 31$. It can be shown that any other valid way to eat exactly 7 portions of ice cream costs more.

D. Teams

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

There are n table bowling players, the rating of the i-th player equals r_i. Compose a maximum number of teams in a such way that:

- each player belongs to at most one team;
- each team contains exactly $a+b$ players;
- each team contains a group of a players with the same rating and a group of b players with another same rating, which must be k times larger than the rating of the players in the first group.

For example, if $n = 12$, $r = [1, 1, 2, 2, 2, 2, 2, 3, 3, 4, 6, 6]$, $a = 1$, $b = 2$, $k = 2$, you can compose two teams with ratings $[1, 2, 2]$ and one team with ratings $[3, 6, 6]$. So, the maximum number of teams is 3.

Find the maximum number of teams by given $n, r_1 \dots r_n, a, b$ and k to compose in respect to the given requirements.

Input

The first line of the input contains four integers n, a, b and k ($1 \le n, a, b \le 3 \cdot 10^5$, $2 \le k \le 1000$). The second line contains the sequence of player's ratings — integers $r_1, r_2, \dots, r_n$ ($1 \le r_i \le 10^6$).

Output

Print only one integer — the maximum number of teams that can be composed from n given players.

Examples

input

```
12 1 2 2
1 1 2 2 2 2 2 3 3 4 6 6
```

output

```
3
```

input

```
14 1 1 3
3 3 1 1 9 9 2 3 6 6 3 18 3 18
```

output

```
6
```

input

```
1 2 3 10
1000000
```

output

```
0
```

E. Double Permutation Inc.

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Polycarp recently became an employee of the company "Double Permutation Inc." Now he is a fan of permutations and is looking for them everywhere!

A permutation in this problem is a sequence of integers $p_1, p_2, \dots, p_k$ such that every integer from 1 to k occurs exactly once in it. For example, the following sequences are permutations of $[3, 1, 4, 2]$, $[1]$ and $[6, 1, 2, 3, 5, 4]$. The following sequences are not permutations: $[0, 1]$, $[1, 2, 2]$, $[1, 2, 4]$ and $[2, 3]$.

In the lobby of the company's headquarter statistics on visits to the company's website for the last n days are published — the sequence $a_1, a_2, \dots, a_n$. Polycarp wants to color all the elements of this sequence in one of three colors (red, green or blue) so that:

- all red numbers, being written out of $a_1, a_2, \dots, a_n$ from left to right (that is, without changing their relative order), must form some permutation (let's call it P);
- all green numbers, being written out of $a_1, a_2, \dots, a_n$ from left to right (that is, without changing their relative order), must form the same permutation P;
- among blue numbers there should not be elements that are equal to some element of the permutation P.

Help Polycarp to color all n numbers so that the total number of red and green elements is maximum.

Input

The first line contains an integer n ($1 \le n \le 2 \cdot 10^5$) — the length of the sequence a. The second line contains n integers $a_1, a_2, \dots, a_n$ ($1 \le a_i \le 2 \cdot 10^5$).

Output

Print a string s of length n such that:

- s_i='R', if the element a_i must be colored in red;
- s_i='G', if the element a_i must be colored in green;
- s_i='B', if the element a_i must be colored in blue.

The string s should maximize the total number of red and green elements when fulfilling the requirements from the main part of the problem statement. If there are several optimal answers, print any of them.

Examples

input

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

output

```
RBBBG
```

input

```
3
1 1 1
```

output

```
BBB
```

input

```
10
3 3 2 2 5 4 1 5 4 1
```

output

```
RGRGRRRGGG
```

input

```
10
3 9 3 1 2 1 2 4 4 4
```

output

```
RBGRRGGBBB
```

F. kotlinkotlinkotlinkotlin...

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Polycarp really likes writing the word "`kotlin`". He wrote this word several times in a row without spaces. For example, he could write the string like "`kotlinkotlinkotlinkotlin`".

Polycarp sliced (cut) the written string into n pieces and mixed them. As a result, he has n strings $s_1, s_2, \dots, s_n$ and he can arrange them in the right order, concatenate (join) all of them and get a string like "`kotlinkotlin...kotlin`".

Help Polycarp to find the right order of strings $s_1, s_2, \dots, s_n$, so that if he writes the strings in this order, he will get the word "`kotlin`" or the sequence of this word.

Pay attention that you must use all given strings and you must use each string only once.

Input

The first line of the input contains one integer n ($1 \le n \le 10^5$) — the number of Polycarp's strings. Next lines of the input contain n Polycarp's strings.

Total sum of their lengths doesn't exceed $3 \cdot 10^5$. It's guaranteed that there is the right order of arrangement the strings that if you concatenate them into one string, you will get some non-empty sequence of the word "`kotlin`".

Output

Print n different integers $p_1, p_2, \dots, p_n$ ($1 \le p_i \le n$), where p_i is an index of the string that should be the i-th in a required concatenation. In other words, the result of concatenation $s_{p_1} + s_{p_2} + \dots + s_{p_n}$ must be in the form "`kotlinkotlin...kotlin`". If there are many solutions, print any of them.

Examples

input

```
2
lin
kot
```

output

```
2 1
```

input

```
4
linkotlinkotlinkotl
kotlin
in
kot
```

output

```
2 4 1 3
```

input

```
8
i
n
tlin
o
ko
t
k
l
```

output

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

G. King's Path

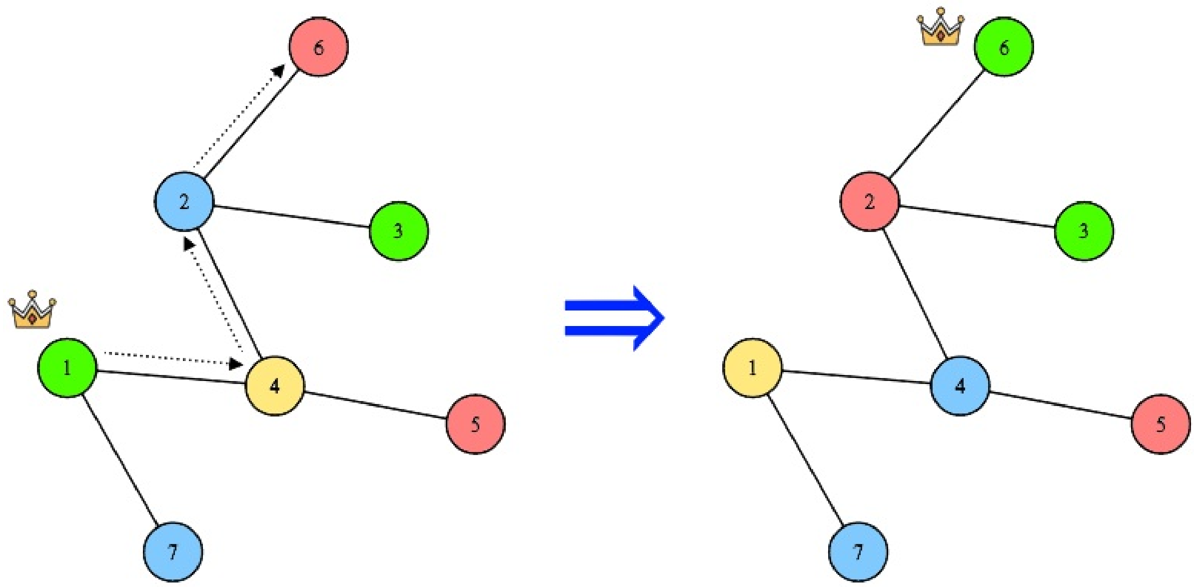
time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

There are n cities and $n-1$ two-way roads in Treeland. Each road connects a pair of different cities. From any city you can drive to any other, moving only along the roads. Cities are numbered from 1 to n. Yes, of course, you recognized an undirected tree in this description.

There is exactly one flag in each city, in the i-th city the flag color is c_i. The colors of the flags in different cities may be the same.

If the King travels along the route $[u_1, u_2, u_3, \ldots, u_k]$, then this means that he starts in the city u_1, then moves to the city u_2 (u_2 is connected by road with u_1), then from u_2 to u_3 (u_3 is connected by road to u_2), and so on until he arrives in the city of u_k. It is possible that during this route the King will visit the same city more than once. In other words, the route $[u_1, u_2, u_3, \ldots, u_k]$ does not necessarily consist of different cities. In terms of graph theory — the King moves from u_1 to u_k along some path $[u_1, u_2, u_3, \ldots, u_k]$, which is not necessarily simple (for all j from 1 to $k-1$ of the city u_j and u_{j+1} are connected by road).

When the King moves from one city to another, city heads exchange flags as a sign of their friendship.



Example of moving the King along the route $[1, 4, 2, 6]$. The color of the vertex matches the color of the flag at that vertex.

For aesthetic reasons, the King wants the flag color in the city i to be equal to d_i for all i from 1 to n. Determine whether the King can choose some route and drive along it so that for each city the flag color in it turns out to be equal to the desired color d_i. Note that the King can choose (and drive) exactly one route. If yes, find **the shortest** possible route for the King.

If the initial colors of the flags already match the King's requirements (i.e. $c_i = d_i$ for all i), then consider that the King makes a route of length $k = 0$.

Input

The first line contains an integer t ($1 \le t \le 10^5$) — the number of test cases to solve. The following are the cases.

Each case begins with a line containing an integer n ($2 \le n \le 2 \cdot 10^5$) — the number of cities in Treeland.

The following is a line of n integers $c_1, c_2, \ldots, c_n$ ($1 \le c_i \le 10^6$), where c_i denotes the color of the flag at the i-th vertex before the King's journey.

The following is a line of n integers $d_1, d_2, \ldots, d_n$ ($1 \le d_i \le 10^6$), where d_i denotes the required flag color at the i-th vertex after the completion of the King's journey.

Further, in the $n-1$ line, the Treeland's roads are listed. Each road is given by a line containing two integers x_j, y_j ($1 \le x_j, y_j \le n$) — numbers of cities that are connected by the j th road.

It is guaranteed that from every city you can get to any other by road (in other words, the system of cities and roads forms an undirected tree).

The sum of all n values for all cases in one test does not exceed $2 \cdot 10^5$.

Output

Print the answers to all cases in the order of their appearance in the input data.

Each answer must begin with a line containing "Yes" (in the case of a positive answer) or "No" (in the case that the required route does not exist). In the case of a positive answer, the following line must contain an integer k — the number of cities in the shortest possible route of the King. The next line should contain the required route $u_1, u_2, \ldots, u_k$ ($1 \le u_i \le n$). You can skip the line if $k = 0$.

Examples

input

```
1
7
2 3 2 7 1 1 3
7 1 2 3 1 2 3
1 7
4 1
2 6
2 3
2 4
```

```
5 4
```

output

```
Yes
4
1 4 2 6
```

input

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

output

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

input

```
3
4
10 20 10 20
20 10 20 10
1 2
1 3
1 4
2
1000000 1000000
1000000 1000000
1 2
10
4 2 2 4 2 4 1 2 3 4
4 2 4 4 3 2 1 2 4 2
5 8
6 9
10 5
1 10
7 10
3 4
5 9
3 10
2 4
```

output

```
No
Yes
0
Yes
5
3 10 5 9 6
```

H. Road Repair in Treeland

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

There are n cities and $n-1$ two-way roads in Treeland. Each road connects a pair of different cities. From any city you can drive to any other, moving only along the roads. Cities are numbered from 1 to n. Yes, of course, you recognized an undirected tree in this description.

The government plans to repair all the roads. Each road will be repaired by some private company. In total, the country has 10^6 private companies that are numbered from 1 to 10^6. It is possible that some companies will not repair roads at all, and some will repair many roads.

To simplify the control over the work of private companies, the following restriction was introduced: for each city, we calculate the number of different companies that repair roads that have this city at one end. This number for each city should not exceed 2. In other words, for each city, there should be no more than two different companies that repair roads related to this city.

The National Anti-Corruption Committee of Treeland raises concerns that all (or almost all) of the work will go to one private company. For this reason, the committee requires that roads be distributed among companies in such a way as to minimize the value of r. For each company, we calculate the number of roads assigned to it, the maximum among all companies is called the number r.

Help the government find such a way to distribute all roads among companies in the required way.

Input

The first line contains an integer t $(1 \leq t \leq 1000)$ — the number of input cases in the input. Next, the input cases themselves are given. The first line of each test case set contains an integer n ($2 \leq n \leq 3000$) — the number of cities in Treeland. Next, in the $n-1$ lines the roads are written: a line contains two integers x_i, y_i ($1 \leq x_i, y_i \leq n$), indicating that the i-th road connects the cities x_i and y_i.

It is guaranteed that the sum of all values n for all test cases in the input does not exceed 3000.

Output

Print the answers for all t test cases in the input. Each test case must begin with a line that contains r — the minimum possible number of roads assigned to the most used company. Next, in the next line print $n-1$ the number $c_1, c_2, \dots, c_{n-1}$ ($1 \leq c_i \leq 10^6$), where c_i indicates the company to repair the i-th road. If there are several optimal assignments, print any of them.

Example

input

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

output

```
1
10 20
3
1 1 1 2 2
2
11 11 12 13 12 13
```

I. Unusual Graph

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Ivan on his birthday was presented with array of non-negative integers $a_1, a_2, \dots, a_n$. He immediately noted that all a_i satisfy the condition $0 \leq a_i \leq 15$.

Ivan likes graph theory very much, so he decided to transform his sequence to the graph.

There will be n vertices in his graph, and vertices u and v will present in the graph if and only if binary notations of integers a_u and a_v are differ in exactly one bit (in other words, $a_u \oplus a_v = 2^k$ for some integer $k \geq 0$. Where $\oplus$ is Bitwise XOR).

A terrible thing happened in a couple of days, Ivan forgot his sequence a, and all that he remembers is constructed graph!

Can you help him, and find any sequence $a_1, a_2, \dots, a_n$, such that graph constructed by the same rules that Ivan used will be the same as his graph?

Input

The first line of input contain two integers n, m ($1 \leq n \leq 500, 0 \leq m \leq \frac{n(n-1)}{2}$): number of vertices and edges in Ivan's graph.

Next m lines contain the description of edges: i-th line contain two integers u_i, v_i ($1 \leq u_i, v_i \leq n; u_i \neq v_i$), describing undirected edge connecting vertices u_i and v_i in the graph.

It is guaranteed that there are no multiple edges in the graph. It is guaranteed that there exists some solution for the given graph.

Output

Output n space-separated integers, $a_1, a_2, \dots, a_n$ ($0 \leq a_i \leq 15$).

Printed numbers should satisfy the constraints: edge between vertices u and v present in the graph if and only if $a_u \oplus a_v = 2^k$ for some integer $k \geq 0$.

It is guaranteed that there exists some solution for the given graph. If there are multiple possible solutions, you can output any.

Examples

input

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

```
3 4
4 1
```

output

```
0 1 0 1
```

input

```
3 0
```

output

```
0 0 0
```