Codeforces Round #411 (Div. 1)

A. Find Amir

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

A few years ago Sajjad left his school and register to another one due to security reasons. Now he wishes to find Amir, one of his schoolmates and good friends.

There are n schools numerated from 1 to n. One can travel between each pair of them, to do so, he needs to buy a ticket. The ticker between schools i and j costs and can be used multiple times. Help Sajjad to find the minimum cost he needs to pay for tickets to visit all schools. He can start and finish in any school.

Input

The first line contains a single integer n $(1 \le n \le 10^5)$ — the number of schools.

Output

Print single integer: the minimum cost of tickets needed to visit all schools.

Examples

input

```
2
```

output

```
0
```

input

```
10
```

output

```
4
```

Note

In the first example we can buy a ticket between the schools that costs .

B. Minimum number of steps

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

We have a string of letters 'a' and 'b'. We want to perform some operations on it. On each step we choose one of substrings "ab" in the string and replace it with the string "bba". If we have no "ab" as a substring, our job is done. Print the minimum number of steps we should perform to make our job done modulo $10^9 + 7$.

The string "ab" appears as a substring if there is a letter 'b' right after the letter 'a' somewhere in the string.

Input

The first line contains the initial string consisting of letters 'a' and 'b' only with length from 1 to 10^6.

Output

Print the minimum number of steps modulo $10^9 + 7$.

Examples

input
ab
output
1

input
aab
output
3

Note

The first example: "ab" $\to$ "bba".

The second example: "aab" $\to$ "abba" $\to$ "bbaba" $\to$ "bbbbaa".

C. Ice cream coloring

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Isart and Modsart were trying to solve an interesting problem when suddenly Kasra arrived. Breathless, he asked: "Can you solve a problem I'm stuck at all day?"

We have a tree T with n vertices and m types of ice cream numerated from 1 to m. Each vertex i has a set of s_i types of ice cream. Vertices which have the i-th $(1 \le i \le m)$ type of ice cream form a connected subgraph. We build a new graph G with m vertices. We put an edge between the v-th and the u-th $(1 \le u, v \le m, u \ne v)$ vertices in G if and only if there exists a vertex in T that has both the v-th and the u-th types of ice cream in its set. The problem is to paint the vertices of G with minimum possible number of colors in a way that no adjacent vertices have the same color.

Please note that we consider that empty set of vertices form a connected subgraph in this problem.

As usual, Modsart don't like to abandon the previous problem, so Isart wants you to solve the new problem.

Input

The first line contains two integer n and m $(1 \le n, m \le 3 \cdot 10^5)$ — the number of vertices in T and the number of ice cream types.

n lines follow, the i-th of these lines contain single integer s_i $(0 \le s_i \le 3 \cdot 10^5)$ and then s_i distinct integers, each between 1 and m — the types of ice cream in the i-th vertex. The sum of s_i doesn't exceed $5 \cdot 10^5$.

$n - 1$ lines follow. Each of these lines describes an edge of the tree with two integers u and v $(1 \le u, v \le n)$ — the indexes of connected by this edge vertices.

Output

Print single integer c in the first line — the minimum number of colors to paint the vertices in graph G.

In the second line print m integers, the i-th of which should be the color of the i-th vertex. The colors should be between 1 and c. If there are some answers, print any of them.

Examples

input

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

output

```
2
1 1 2
```

input

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

output

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

Note

In the first example the first type of ice cream is present in the first vertex only, so we can color it in any color. The second and the third ice cream are both presented in the second vertex, so we should paint them in different colors.

In the second example the colors of the second, the fourth and the fifth ice cream should obviously be distinct.

D. Expected diameter of a tree

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

Pasha is a good student and one of MoJaK's best friends. He always have a problem to think about. Today they had a talk about the following problem.

We have a forest (acyclic undirected graph) with n vertices and m edges. There are q queries we should answer. In each query two vertices v and u are given. Let V be the set of vertices in the connected component of the graph that contains v, and U be the set of vertices in the connected component of the graph that contains u. Let's add an edge between some vertex and some vertex in and compute the value d of the resulting component. If the resulting component is a tree, the value d is the *diameter* of the component, and it is equal to `-1` otherwise. What is the expected value of d, if we choose vertices a and b from the sets uniformly at random?

Can you help Pasha to solve this problem?

The *diameter* of the component is the maximum *distance* among some pair of vertices in the component. The *distance* between two vertices is the minimum number of edges on some path between the two vertices.

Note that queries don't add edges to the initial forest.

Input

The first line contains three integers n, m and $q(1 \le n, m, q \le 10^5)$ — the number of vertices, the number of edges in the graph and the number of queries.

Each of the next m lines contains two integers u_i and v_i $(1 \le u_i, v_i \le n)$, that means there is an edge between vertices u_i and v_i.

It is guaranteed that the given graph is a forest.

Each of the next q lines contains two integers u_i and v_i $(1 \le u_i, v_i \le n)$ — the vertices given in the i-th query.

Output

For each query print the expected value of d as described in the problem statement.

Your answer will be considered correct if its absolute or relative error does not exceed 10^{-6}. Let's assume that your answer is a, and the jury's answer is b. The checker program will consider your answer correct, if .

Examples

input

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

output

```
-1
2.0000000000
```

input

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

output

```
-1
2.6666666667
2.6666666667
```

Note

In the first example the vertices 1 and 3 are in the same component, so the answer for the first query is `-1`. For the second query there are two options to add the edge: one option is to add the edge $1-2$, the other one is $2-3$. In both ways the resulting diameter is 2, so the answer is 2.

In the second example the answer for the first query is obviously `-1`. The answer for the second query is the average of three cases: for added edges $1-2$ or $1-3$ the diameter is 3, and for added edge $1-4$ the diameter is 2. Thus, the answer is .

E. The same permutation

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Seyyed and MoJaK are friends of Sajjad. Sajjad likes a permutation. Seyyed wants to change the permutation in a way that Sajjad won't like it. Seyyed thinks more swaps yield more probability to do that, so he makes MoJaK to perform a swap between every pair of positions (i, j), where $i < j$, exactly once. MoJaK doesn't like to upset Sajjad.

Given the permutation, determine whether it is possible to swap all pairs of positions so that the permutation stays the same. If it is possible find how to do that.

Input

The first line contains single integer n ($1 \le n \le 1000$) — the size of the permutation.

As the permutation is not important, you can consider $a_i = i$, where the permutation is $a_1, a_2, ..., a_n$.

Output

If it is not possible to swap all pairs of positions so that the permutation stays the same, print "`NO`",

Otherwise print "`YES`", then print lines: the i-th of these lines should contain two integers a and b ($a < b$) — the positions where the i-th swap is performed.

Examples

input

```
3
```

output

```
NO
```

input

```
1
```

output

```
YES
```

F. Fake bullions

time limit per test: 3 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

In Isart people don't die. There are n gangs of criminals. The i-th gang contains s_i evil people numerated from 0 to $s_i - 1$. Some of these people took part in a big mine robbery and picked **one** gold bullion each (these people are given in the input). That happened 10^{100} years ago and then all of the gangs escaped to a remote area, far from towns.

During the years, they were copying some gold bullions according to an organized plan in order to not get arrested. They constructed a **tournament** directed graph (a graph where there is exactly one directed edge between every pair of vertices) of gangs (the graph is given in the input). In this graph an edge from u to v means that in the i-th hour the person of the gang u can send a fake gold bullion to person of gang v. He sends it if he has some bullion (real or fake), while the receiver doesn't have any. Thus, at any moment each of the gangsters has zero or one gold bullion. Some of them have real bullions, and some of them have fake ones.

In the beginning of this year, the police has finally found the gangs, but they couldn't catch them, as usual. The police decided to open a jewelry store so that the gangsters would sell the bullions. Thus, every gangster that has a bullion (fake or real) will try to sell it. If he has a real gold bullion, he sells it without problems, but if he has a fake one, there is a choice of two events that can happen:

- The person sells the gold bullion successfully.
- The person is arrested by police.

The power of a gang is the number of people in it that successfully sold their bullion. After all selling is done, the police arrests b gangs out of *top* gangs. A gang with power p is in *top* gangs if there are no more than $a - 1$ gangs with power strictly more than p. Consider all possible results of selling fake gold bullions and all possible choice of b gangs among the *top* gangs. Count the number of different sets of these b gangs modulo $10^9 + 7$. Two sets X and Y are considered different if some gang is in X and isn't in Y.

Input

The first line contains four integers n, a and b $(1 \le b \le a \le n \le 5 \cdot 10^3)$ — the number of gangs, the constants a and b from the statement.

Then n lines follow, each line contains a string of size n consisting of zeros and ones. The j-th character in the i-th of these lines is equal to 1, then the vertex i have a directed edge to the vertex j. It is guaranteed that $a_{ii} = 0$ and $a_{ij} + a_{ji} = 1$ if $i \neq j$.

Then n lines follow, each line starts with the integer s_i $(1 \le s_i \le 2 \cdot 10^6)$ — the number of gangsters in the i-th gang, and then contains a string of zeros and ones with length s_i. The j-th character is 0 if the j-th person of the i-th gang had a real gold bullion initially, otherwise it is 1. It is guaranteed that the sum of s_i does not exceed $2 \cdot 10^6$.

Output

Print single integer: the number of different sets of b gangs the police can arrest modulo $10^9 + 7$.

Examples

input

```
2 2 1
01
00
5 11000
6 100000
```

output

```
2
```

input

```
5 2 1
00000
10000
11011
11000
11010
2 00
1 1
6 100110
1 0
1 0
```

output

```
5
```