Codeforces β
Sponsored by Telegram

Codeforces Round #406 (Div. 1)

A. Berzerk

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Rick and Morty are playing their own version of Berzerk (which has nothing in common with the famous Berzerk game). This game needs a huge space, so they play it with a computer.

In this game there are n objects numbered from 1 to n arranged in a circle (in clockwise order). Object number 1 is a black hole and the others are planets. There's a monster in one of the planet. Rick and Morty don't know on which one yet, only that he's not initially in the black hole, but Unity will inform them before the game starts. But for now, they want to be prepared for every possible scenario.

Each one of them has a set of numbers between 1 and $n - 1$ (inclusive). Rick's set is s_1 with k_1 elements and Morty's is s_2 with k_2 elements. One of them goes first and the player changes alternatively. In each player's turn, he should choose an arbitrary number like x from his set and the monster will move to his x-th next object from its current position (clockwise). If after his move the monster gets to the black hole he wins.

Your task is that for each of monster's initial positions and who plays first determine if the starter wins, loses, or the game will stuck in an infinite loop. In case when player can lose or make game infinity, it more profitable to choose infinity game.

Input

The first line of input contains a single integer n $(2 \le n \le 7000)$ — number of objects in game.

The second line contains integer k_1 followed by k_1 distinct integers $s_{1,1}, s_{1,2}, \ldots, s_{1,k_1}$ — Rick's set.

The third line contains integer k_2 followed by k_2 distinct integers $s_{2,1}, s_{2,2}, \ldots, s_{2,k_2}$ — Morty's set

$1 \le k_i \le n - 1$ and $1 \le s_{i,1}, s_{i,2}, \ldots, s_{i,k_i} \le n - 1$ for $1 \le i \le 2$.

Output

In the first line print $n - 1$ words separated by spaces where i-th word is "Win" (without quotations) if in the scenario that Rick plays first and monster is initially in object number $i + 1$ he wins, "Lose" if he loses and "Loop" if the game will never end.

Similarly, in the second line print $n - 1$ words separated by spaces where i-th word is "Win" (without quotations) if in the scenario that Morty plays first and monster is initially in object number $i + 1$ he wins, "Lose" if he loses and "Loop" if the game will never end.

Examples

input

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

output

```
Lose Win Win Loop
Loop Win Win Win
```

input

```
8
4 6 2 3 4
2 3 6
```

output

```
Win Win Win Win Win Win Win
Lose Win Lose Lose Win Lose Lose
```

B. Legacy

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

Rick and his co-workers have made a new radioactive formula and a lot of bad guys are after them. So Rick wants to give his legacy to Morty before bad guys catch them.

There are n planets in their universe numbered from 1 to n. Rick is in planet number s (the earth) and he doesn't know where Morty is. As we all know, Rick owns a portal gun. With this gun he can open one-way portal from a planet he is in to any other planet (including that planet). But there are limits on this gun because he's still using its free trial.

By default he can not open any portal by this gun. There are q plans in the website that sells these guns. Every time you purchase a plan you can only use it once but you can purchase it again if you want to use it more.

Plans on the website have three types:

1. With a plan of this type you can open a portal from planet v to planet u.
2. With a plan of this type you can open a portal from planet v to any planet with index in range $[l, r]$.
3. With a plan of this type you can open a portal from any planet with index in range $[l, r]$ to planet v.

Rick doesn't known where Morty is, but Unity is going to inform him and he wants to be prepared for when he finds and start his journey immediately. So for each planet (including earth itself) he wants to know the minimum amount of money he needs to get from earth to that planet.

Input

The first line of input contains three integers n, q and s ($1 \le n, q \le 10^5$, $1 \le s \le n$) — number of planets, number of plans and index of earth respectively.

The next q lines contain the plans. Each line starts with a number t, type of that plan ($1 \le t \le 3$). If $t = 1$ then it is followed by three integers v, u and w where w is the cost of that plan ($1 \le v, u \le n$, $1 \le w \le 10^9$). Otherwise it is followed by four integers v, l, r and w where w is the cost of that plan ($1 \le v \le n$, $1 \le l \le r \le n$, $1 \le w \le 10^9$).

Output

In the first and only line of output print n integers separated by spaces. i-th of them should be minimum money to get from earth to i-th planet, or -1 if it's impossible to get to that planet.

Examples

input

```
3 5 1
2 3 2 3 17
2 3 2 2 16
2 2 2 3 3
3 3 1 1 12
1 3 3 17
```

output

```
0 28 12
```

input

```
4 3 1
3 4 1 3 12
2 2 3 4 10
1 2 4 16
```

output

```
0 -1 -1 12
```

Note

In the first sample testcase, Rick can purchase 4th plan once and then 2nd plan in order to get to get to planet number 2.

C. Till I Collapse

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

Rick and Morty want to find MR. PBH and they can't do it alone. So they need of Mr. Meeseeks. They Have generated n Mr. Meeseeks, standing in a line numbered from 1 to n. Each of them has his own color. i-th Mr. Meeseeks' color is a_i.

Rick and Morty are gathering their army and they want to divide Mr. Meeseeks into some squads. They don't want their squads to be too colorful, so each squad should have Mr. Meeseeks of at most k different colors. Also each squad should be a continuous subarray of Mr. Meeseeks in the line. Meaning that for each $1 \le i \le e \le j \le n$, if Mr. Meeseeks number i and Mr. Meeseeks number j are in the same squad then Mr. Meeseeks number e should be in that same squad.

Also, each squad needs its own presidio, and building a presidio needs money, so they want the total number of squads to be minimized.

Rick and Morty haven't finalized the exact value of k, so in order to choose it, for each k between 1 and n (inclusive) need to know the minimum number of presidios needed.

Input

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

The second line contains n integers $a_1, a_2, ..., a_n$ separated by spaces $(1 \le a_i \le n)$ — colors of Mr. Meeseeks in order they standing in a line.

Output

In the first and only line of input print n integers separated by spaces. i-th integer should be the minimum number of presidios needed if the value of k is i.

Examples

input

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

output

```
4 2 1 1 1
```

input

```
8
1 5 7 8 1 7 6 1
```

output

```
8 4 3 2 1 1 1 1
```

Note

For the first sample testcase, some optimal ways of dividing army into squads for each k are:

1. $[1], [3], [4], [3, 3]$
2. $[1], [3, 4, 3, 3]$
3. $[1, 3, 4, 3, 3]$
4. $[1, 3, 4, 3, 3]$
5. $[1, 3, 4, 3, 3]$

For the second testcase, some optimal ways of dividing army into squads for each k are:

1. $[1], [5], [7], [8], [1], [7], [6], [1]$
2. $[1, 5], [7, 8], [1, 7], [6, 1]$
3. $[1, 5, 7], [8], [1, 7, 6, 1]$
4. $[1, 5, 7, 8], [1, 7, 6, 1]$
5. $[1, 5, 7, 8, 1, 7, 6, 1]$
6. $[1, 5, 7, 8, 1, 7, 6, 1]$
7. $[1, 5, 7, 8, 1, 7, 6, 1]$
8. $[1, 5, 7, 8, 1, 7, 6, 1]$

D. Rap God

time limit per test: 7 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Rick is in love with Unity. But Mr. Meeseeks also love Unity, so Rick and Mr. Meeseeks are "love rivals".

Unity loves rap, so it decided that they have to compete in a rap game (battle) in order to choose the best. Rick is too nerds, so instead he's gonna make his verse with running his original algorithm on lyrics "Rap God" song.

His algorithm is a little bit complicated. He's made a tree with n vertices numbered from 1 to n and there's a lowercase english letter written on each edge. He denotes $str(a, b)$ to be the string made by writing characters on edges on the shortest path from a to b one by one (a string of length equal to distance of a to b). Note that $str(a, b)$ is reverse of $str(b, a)$ and $str(a, a)$ is empty.

In order to make the best verse he can, he needs to answer some queries, but he's not a computer scientist and is not able to answer those queries, so he asked you to help him. Each query is characterized by two vertices x and y ($x \neq y$). Answer to this query is the number of vertices like z such that $z \neq x, z \neq y$ and $str(x, y)$ is lexicographically larger than $str(x, z)$.

String $x = x_1x_2...x_{|x|}$ is lexicographically larger than string $y = y_1y_2...y_{|y|}$, if either $|x| > |y|$ and $x_1 = y_1, x_2 = y_2, ..., x_{|y|} = y_{|y|}$, or exists such number r ($r < |x|, r < |y|$), that $x_1 = y_1, x_2 = y_2, ..., x_r = y_r$ and $x_{r+1} > y_{r+1}$. Characters are compared like their ASCII codes (or alphabetic order).

Help Rick get the girl (or whatever gender Unity has).

Input

The first line of input contain two integers n and q ($2 \leq n \leq 20000$, $1 \leq q \leq 20000$) — number of vertices in tree and number of queries respectively.

The next n - 1 lines contain the edges. Each line contains two integers v and u (endpoints of the edge) followed by an English lowercase letter c ($1 \leq v, u \leq n, v \neq u$).

The next q line contain the queries. Each line contains two integers x and y ($1 \leq x, y \leq n, x \neq y$).

Output

Print the answer for each query in one line.

Examples

input

```
4 3
4 1 t
3 2 p
1 2 s
3 2
1 3
2 1
```

output

```
0
1
1
```

input

```
8 4
4 6 p
3 7 o
7 8 p
4 5 d
1 3 o
4 3 p
3 2 e
8 6
3 7
8 1
4 3
```

output

```
6
1
3
1
```

Note

Here's the tree of first sample testcase:

Here's the tree of second sample testcase:

In this test:

- $str(8, 1)$ = `poo`
- $str(8, 2)$ = `poe`
- $str(8, 3)$ = `po`
- $str(8, 4)$ = `pop`
- $str(8, 5)$ = `popd`
- $str(8, 6)$ = `popp`
- $str(8, 7)$ = `p`

So, for the first query, and for the third query is the answer.

E. ALT

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

ALT is a planet in a galaxy called "Encore". Humans rule this planet but for some reason there's no dog in their planet, so the people there are sad and depressed. Rick and Morty are universal philanthropists and they want to make people in ALT happy.

ALT has n cities numbered from 1 to n and $n - 1$ bidirectional roads numbered from 1 to $n - 1$. One can go from any city to any other city using these roads.

There are two types of people in ALT:

1. Guardians. A guardian lives in a house alongside a road and guards the road.
2. Citizens. A citizen lives in a house inside a city and works in an office in another city.

Every person on ALT is either a guardian or a citizen and there's exactly one guardian alongside each road.

Rick and Morty talked to all the people in ALT, and here's what they got:

- There are m citizens living in ALT.
- Citizen number i lives in city number x_i and works in city number y_i.
- Every day each citizen will go through all roads along the shortest path from his home to his work.
- A citizen will be happy if and only if either he himself has a puppy himself or all of guardians along his path to his work has a puppy (he sees the guardian's puppy in each road and will be happy).
- A guardian is always happy.

You need to tell Rick and Morty the minimum number of puppies they need in order to make all people in ALT happy, and also provide an optimal way to distribute these puppies.

Input

The first line of input contains two integers n and m ($2 \le n \le 2 \times 10^4$, $1 \le m \le 10^4$) — number of cities and number of citizens respectively.

The next $n - 1$ lines contain the roads, i-th line contains endpoint of i-th edge, v and u ($1 \le v, u \le n$, $v \ne u$).

The next m lines contain the information about citizens. i-th line contains two integers x_i and y_i ($1 \le x_i, y_i \le n$, $x_i \ne y_i$).

Output

In the first line of input print a single integer k, the total number of puppies they need ($1 \le k \le n$).

In the second line print an integer q, the number of puppies to give to citizens, followed by q distinct integers $a_1, a_2, ..., a_q$, index of citizens to give puppy to ($0 \le q \le min(m, k)$, $1 \le a_i \le m$).

In the third line print an integer e, the number of puppies to give to guardians, followed by e distinct integers $b_1, b_2, ..., b_e$, index of road of guardians to give puppy to ($0 \le e \le min(n - 1, k)$, $1 \le b_i \le n - 1$).

Sum of q and e should be equal to k.

Examples

input

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

output

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

input

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

```
4 2
```

output

```
3
1 6
2 2 3
```

Note

Map of ALT in the first sample testcase (numbers written on a road is its index):

Map of ALT in the second sample testcase (numbers written on a road is its index):