Codeforces Round #406 (Div. 2)

A. The Monster

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

A monster is chasing after Rick and Morty on another planet. They're so frightened that sometimes they scream. More accurately, Rick screams at times $b, b + a, b + 2a, b + 3a, ...$ and Morty screams at times $d, d + c, d + 2c, d + 3c, ...$

The Monster will catch them if at any point they scream at the same time, so it wants to know when it will catch them (the first time they scream at the same time) or that they will never scream at the same time.

Input

The first line of input contains two integers a and b $(1 \leq a, b \leq 100)$.

The second line contains two integers c and d $(1 \leq c, d \leq 100)$.

Output

Print the first time Rick and Morty will scream at the same time, or -1 if they will never scream at the same time.

Examples

input
20 2 9 19
output
82

input
2 1 16 12
output
-1

Note

In the first sample testcase, Rick's 5th scream and Morty's 8th time are at time 82.

In the second sample testcase, all Rick's screams will be at odd times and Morty's will be at even times, so they will never scream at the same time.

B. Not Afraid

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

Since the giant heads have appeared in the sky all humanity is in danger, so **all** Ricks and Mortys from all parallel universes are gathering in groups to find a solution to get rid of them.

There are n parallel universes participating in this event (n Ricks and n Mortys). I. e. each of n universes has one Rick and one Morty. They're gathering in m groups. Each person can be in many groups and a group can contain an arbitrary number of members.

Ricks and Mortys have registered online in these groups. So, a person can have joined a group more than once (developer of this website hadn't considered this possibility).

Summer from universe #1 knows that in each parallel universe (including hers) exactly one of Rick and Morty from that universe is a traitor and is loyal, but no one knows which one. She knows that we are doomed if there's a group such that every member in that group is a traitor (they will plan and destroy the world).

Summer knows that if there's a possibility that world ends (there's a group where all members are traitors) she should immediately cancel this event. So she wants to know if she should cancel the event. You have to tell her yes if and only if there's at least one scenario (among all 2^n possible scenarios, 2 possible scenarios for who a traitor in each universe) such that in that scenario the world will end.

Input

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

The next m lines contain the information about the groups. i-th of them first contains an integer k (number of times someone joined i-th group, $k > 0$) followed by k integers $v_{i,1}, v_{i,2}, ..., v_{i,k}$. If $v_{i,j}$ is negative, it means that Rick from universe number $-v_{i,j}$ has joined this group and otherwise it means that Morty from universe number $v_{i,j}$ has joined it.

Sum of k for all groups does not exceed 10^4.

Output

In a single line print the answer to Summer's question. Print "YES" if she should cancel the event and "NO" otherwise.

Examples

input
4 2 1 -3 4 -2 3 2 -3
output
YES

input
5 2 5 3 -2 1 -1 5 3 -5 2 5
output
NO

input
7 2 3 -1 6 7 7 -5 4 2 4 7 -3 4
output
YES

Note

In the first sample testcase, 1st group only contains the Rick from universe number 3, so in case he's a traitor, then all members of this group are traitors and so Summer should cancel the event.

C. 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 $s_{1,1}, s_{1,2}, ..., s_{1,k_1}$ — Rick's set.

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

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

D. 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.

E. 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. $[1, 5, 7, 8, 1, 7, 6, 1]$