Preliminaries

Official Problemset

전국 대학생 프로그래밍 대회 동아리 연합
여름 대회 2020 예선

주최 전국 대학생 프로그래밍 대회 동아리 연합

주관 고려대학교 SW중심대학사업단

후원 STARTLINK

NAVER D2

MINDs Lab.

ALGOSPOT

KENNYSOFT

SOLVED.AC

문제 목록

문제지에 있는 문제가 총 10문제가 맞는지 확인하시기 바랍니다.

A 수학은 비대면강의입니다

B 길이 문자열

C 삼항 연산자

D ㄷㄷㄷㅈ

E 감자 농장

F 전투 시뮬레이션

G 루머

H 사과나무

I 인버스 ㄷㄷㄷㅈ

J 역학 조사

모든 문제의 메모리 제한은 1GB로 동일합니다.

문제 A. 수학은 비대면강의입니다

시간 제한	1 초
메모리 제한	1024 MB

수현이는 4차 산업혁명 시대에 살고 있는 중학생이다. 코로나 19로 인해, 수현이는 버추얼 학교로 버추얼 출석해 버추얼 강의를 듣고 있다. 수현이의 버추얼 선생님은 문자가 2개인 연립방정식을 해결하는 방법에 대해 강의하고, 다음과 같은 문제를 숙제로 냈다.

다음 연립방정식에서 x와 y의 값을 계산하시오.

$$\begin{cases} ax+by=c \\ dx+ey=f \end{cases}$$

4차 산업혁명 시대에 숙제나 하고 앉아있는 것보다 버추얼 친구들을 만나러 가는 게 더 가치있는 일이라고 생각했던 수현이는 이런 연립방정식을 풀 시간이 없었다. 다행히도, 버추얼 강의의 숙제 제출은 인터넷 창의 빈 칸에 수들을 입력하는 식이다. 각 칸에는 -999 이상 999 이하의 정수만 입력할 수 있다. 수현이가 버추얼 친구들을 만나러 버추얼 세계로 떠날 수 있게 도와주자.

입력

정수 a, b, c, d, e, f가 공백으로 구분되어 차례대로 주어진다. ($-999 \le a,b,c,d,e,f \le 999$)

문제에서 언급한 방정식을 만족하는 (x,y)가 유일하게 존재하고, 이 때 x와 y가 각각 -999 이상 999 이하의 정수인 경우만 입력으로 주어짐이 보장된다.

출력

문제의 답인 x와 y를 공백으로 구분해 출력한다.

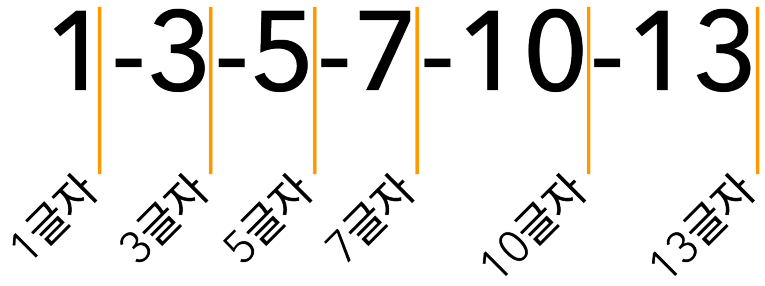
입출력 예시

표준 입력(stdin)	표준 출력(stdout)
1 3 -1 4 1 7	2 -1
2 5 8 3 -4 -11	-1 2

이 페이지는 공백입니다

문제 B. 길이 문자열

시간 제한	3 초
메모리 제한	1024 MB



길이 문자열은 숫자 0–9와 하이픈('-')으로만 이루어진 문자열 중 다음 조건을 만족하는 것을 가리킨다.

- '-'이 2개 이상 연속해서 등장하지 않는다.
- 문자열의 첫 문자는 '0'이 아니다.
- 문자열의 마지막 문자는 '-'이 아니다.
- '-'의 다음 문자로 '0'이 등장하지 않는다.
- 문자열의 숫자로만 이루어진 접미사 중 가장 긴 것을 10진법의 수로 해석하면 문자열의 길이와 같다. 이 때 그러한 접미사가 빈 문자열이면 0으로 해석한다.
- 문자열에 '-'이 등장한다면 문자열의 처음부터 가장 마지막에 등장하는 '-' 앞까지의 부분 문자열이 길이 문자열이다.

임의의 음이 아닌 정수 n에 대해 길이가 n인 길이 문자열은 유일하게 한 개 존재한다. 다음은 각각 길이 5, 8, 13인 길이 문자열의 예시이다.

```
1-3-5
-2-4-6-8
1-3-5-7-10-13
```

자연수 a와 음이 아닌 정수 b가 주어졌을 때 길이가 $a \times 10^b$인 길이 문자열을 찾아보자.

입력

첫째줄에 테스트 케이스의 개수 T가 주어진다. ($1 \le T \le 100\,000$)

각 테스트 케이스마다 두 개의 정수 a, b가 공백으로 구분되어 한 줄에 주어진다. ($1 \le a \le 10^9$, $0 \le b \le 10^6$)

출력

각 테스트 케이스마다 길이 $a \times 10^b$인 길이 문자열을 출력한다. 이때 $a \times 10^b \ge 21$이면 문자열의 맨 앞의 17글자만 출력 예시와 같은 형식으로 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
3 5 0 8 0 13 0	1-3-5 -2-4-6-8 1-3-5-7-10-13
2 25 4 32 6	1-3-5-7-10-13-16-... -2-4-6-8-11-14-17...

문제 C. 삼항 연산자

시간 제한	0.5 초
메모리 제한	1024 MB

N개의 참/거짓 변수를 가진 식이 주어진다. 변수의 값으로 가능한 2^N개의 경우에 대하여 식의 값이 0인 경우의 수를 구하는 프로그램을 작성하여라.

이 문제에서 올바른 식은 아래 BNF 표기법에서 expression을 뜻한다.

- boolean ::= '0' | '1'
- variable ::= 'a' | 'b' | ... | N번째 알파벳 소문자
- value ::= boolean | variable
- condition ::= value '==' value
- expression ::= value | condition '?' expression ':' expression

식의 값은 $eval(\text{expression})$을 의미하며 아래와 같이 재귀적으로 계산된다. 잘 생각해보면 올바른 식이 주어졌을 때 해당 식을 계산하는 방법이 유일하다는 것을 알 수 있다.

- $eval(\text{value}) = \text{value}$
- $eval(\text{condition}) = eval(\text{value1 '==' value2}) = \begin{cases} 1 & \text{if } eval(\text{value1}) = eval(\text{value2}) \\ 0 & \text{otherwise} \end{cases}$
- $eval(\text{expression}) = eval(\text{value}) = \text{value}$
- $eval(\text{expression}) = eval(\text{condition '?' expression1 ':' expression2})$
  $= \begin{cases} eval(\text{expression1}) & \text{if } eval(\text{condition}) = 1 \\ eval(\text{expression2}) & \text{otherwise} \end{cases}$

입력

첫 번째 줄에는 변수의 수 N $(1 \le N \le 26)$이 주어진다.

두 번째 줄에는 식에 해당하는 길이 1 이상 1 000 이하의 문자열이 주어진다. 식은 '0', '1', 'a'–(N번째 알파벳 소문자), '=', '?', ':'로만 구성되며 올바른 식만 주어진다.

출력

식의 값이 0이 되도록 변수의 값을 할당하는 방법의 수를 출력한다.

입출력 예시

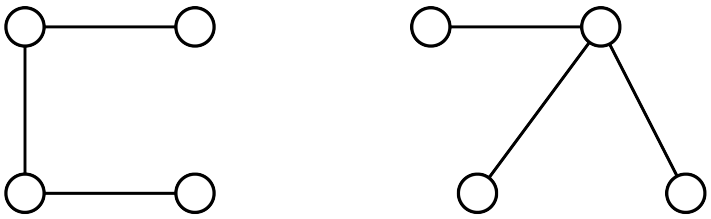
표준 입력(stdin)	표준 출력(stdout)
2 a==b?a:0	3
10 0	1024

이 페이지는 공백입니다

문제 D. ㄷㄷㄷㅈ

시간 제한	2 초
메모리 제한	1024 MB

어느 날, 트리를 물끄러미 보고 있던 동현이는 엄청난 사실을 하나 발견했다. 바로 정점이 네 개인 트리는 'ㄷ'과 'ㅈ'의 두 종류밖에 없다는 사실이다!



정점이 네 개 이상 있는 임의의 트리에 대해, 그 트리에서 정점 네 개로 이루어진 집합을 고르자. 전체 트리의 간선들 중 집합에 속한 두 정점을 잇는 간선만을 남겼을 때, 네 개의 정점이 하나의 트리 형태로 이어지게 된다면 'ㄷ' 모양이거나 'ㅈ' 모양일 것이다. 트리에서 'ㄷ'의 개수와 'ㅈ'의 개수를 각각 트리에서 'ㄷ' 모양, 'ㅈ' 모양을 이루는 정점 네 개짜리 집합의 개수라고 하자.

이제, 동현이는 세상의 모든 트리를 다음과 같은 세 종류로 나누었다.

- D-트리 : 'ㄷ'이 'ㅈ'의 3배보다 많은 트리
- G-트리 : 'ㄷ'이 'ㅈ'의 3배보다 적은 트리
- DUDUDUNGA-트리 : 'ㄷ'이 'ㅈ'의 정확히 3배만큼 있는 트리

신이 난 동현이는 트리만 보이면 그 트리에 있는 'ㄷ'과 'ㅈ'이 몇 개인지 세고 다니기 시작했다. 하지만 곧 정점이 30만 개나 있는 트리가 동현이 앞에 나타났고, 동현이는 그만 정신을 잃고 말았다. 동현이를 대신해 주어진 트리가 D-트리인지 G-트리인지 아니면 DUDUDUNGA-트리인지 알려주자!

입력

첫 번째 줄에 트리의 정점 수 N이 주어진다. $(4 \le N \le 300\,000)$

두 번째 줄부터 $N-1$개의 줄에 트리의 각 간선이 잇는 두 정점의 번호 u, v가 주어진다. $(1 \le u, v \le N)$

출력

첫 번째 줄에 주어진 트리가 D-트리라면 `D`, G-트리라면 `G`, DUDUDUNGA-트리라면 `DUDUDUNGA`를 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
4 1 2 2 3 3 4	D
4 1 2 1 3 1 4	G
6 1 2 2 3 3 4 4 5 4 6	DUDUDUNGA

문제 E. 감자 농장

시간 제한	2 초
메모리 제한	1024 MB

이하는 갊자에게 맛있는 감자튀김을 바치기 위해 감자 농장을 꾸렸다. 이 감자 농장은 1번 칸부터 N번 칸까지 일자로 연속되어 있는 땅으로 볼 수 있다. 1번 칸이 서쪽에 있으며, N번 칸이 동쪽에 있다. 안타깝게도 이하가 산 땅의 일부엔 바위가 박혀있어, 감자를 심을 수도 없을 뿐더러 걸어다니기도 불편했다. 그러나 노력 끝에 이하는 감자를 이 땅에 일부 심었고, 4차 산업혁명의 혁신적인 기술력을 이용하여 절실히 가꾸었다.

마침내 수확의 시간이 다가왔고, 이하는 이제 감자를 수확하려고 한다. 이하는 기계적으로 움직이는 것을 좋아해서, 다음과 같은 규칙을 지키며 움직인다.

- 처음에 이하는 바위도 감자도 없는 칸에서 시작하며, 최초 이동 방향은 동쪽이다.
- 이하는 한 칸씩 서쪽 또는 동쪽으로 이동하며, 매 이동엔 1의 시간이 걸린다.
- 이하가 도달한 칸에 감자가 심어져 있으면, 이하는 땅에 있는 감자를 수확하고 방향을 반대로 전환한다.
  - 감자는 매 칸별로 최대 1개 심어져 있으며, 감자를 수확하는 시간 및 방향 전환 시간은 무시할 정도로 작다.
- 이하가 도달한 칸에 바위가 있으면, 이하는 방향을 반대로 전환한다.

이하는 1번 칸에서 한 칸 서쪽으로 이동하거나, N번 칸에서 한 칸 동쪽으로 이동한 후 감자 농장을 벗어나게 된다.

여기까지가 이하의 계획이었으나, 고민해보니 처음 시작 위치에 따라 수확할 수 있는 감자의 개수와 총 소요 시간이 다르다는 점을 깨달았다. 심지어 농장의 상태에 따라, 몇몇 시작 위치에선 감자 농장을 벗어날 수 없다는 사실도 알아냈다. 그래서 이하는 이 문제를 빠르게 풀고, 더 나은 방법을 구상해보고자 한다. 감자를 수확하고 싶은 이하를 도와주자.

입력

첫 번째 줄에는 두 자연수 N, Q가 공백으로 구분되어 주어진다. ($1 \le N \le 10^6$, $1 \le Q \le \min\left(N, 10^5\right)$) N은 감자 농장의 칸 수, 그리고 Q는 이하의 질의 횟수이다.

두 번째 줄에는 길이 N의 문자열 S가 주어진다. S의 각 문자는 `P`, `R`, 또는 `.`이다. S의 i ($1 \le i \le N$)번째 문자가 `P`이면 i번 칸에 감자가 심어져 있음을 의미하고, `R`이면 바위가 있음을 의미하며, `.`이면 아무 것도 없음을 의미한다.

세 번째 줄부터 Q개의 줄에 걸쳐 이하의 질의가 주어진다. 각 줄은 자연수 x로 이루어져 있다. ($1 \le x \le N$) 이는 이하가 초기 위치를 x번 칸으로 삼았을 때 어떻게 될지 알고 싶음을 의미한다. S의 x번째 글자는 `.`임이 보장되며, 모든 질의는 서로 다르다.

각 질의는 독립적으로 계산해야 한다.

출력

매 질의별로 한 줄씩, 두 정수 p와 t를 공백으로 구분하여 출력한다. p는 이하가 해당 위치에서 시작했을 때 수확한 감자의 개수이다. t는 이하가 감자 농장을 벗어날 수 있다면 걸린 시간이고, 아니면 `-1`이다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
6 3 .P.PR. 1 3 6	1 3 2 11 0 1
3 1 R.R 2	0 -1
11 5 ..RP.RP.P.P 10 1 5 8 2	2 6 0 5 1 -1 3 18 0 4
1 1 . 1	0 1

문제 F. 전투 시뮬레이션

시간 제한	2 초
메모리 제한	1024 MB

당신은 $H \times W$ 크기의 2차원 격자 맵에서 두 세력이 전투를 벌이는 시뮬레이션 게임을 개발하고 있다.

격자의 각 칸은 y좌표와 x좌표의 쌍 (y,x)로 표현할 수 있다. 첫 번째 줄에 있는 칸들은 왼쪽부터 순서대로 $(0,0), (0,1), \cdots, (0,W-1)$로 표현하며, 두 번째 줄의 칸들은 $(1,0), (1,1), \cdots, (1,W-1)$로 표현한다. 같은 방법으로 H번째 줄까지의 모든 칸들을 좌표로 표현한다. 어떤 칸의 위, 아래, 왼쪽, 오른쪽 칸들은 그 칸에 '인접한' 칸이라고 한다.

맵은 다양한 지형으로 이루어져 있고 각 지형은 정해진 '**험준도**' 수치를 가진다. 또, 맵에는 여러 유닛이 서로 겹치지 않게 배치되어 있으며, 각 유닛은 전투 중인 두 세력 중 한 세력에 속한다. 각 유닛은 맵 밖으로 벗어나지 않는 한 위치한 곳으로부터 인접한 칸으로 이동할 수 있다. 이동 시에는 해당 칸의 지형이 가진 험준도만큼의 스태미나를 소모하게 된다. 일부 지형은 너무 험준하여 이동이 불가능할 수도 있다. 세력이 다른 두 유닛이 인접해 있다면 그 둘은 교전 상태이다.

모든 유닛은 전투식량을 든든하게 챙겨 먹고 나왔기 때문에 무한한 스태미나를 가지고 있다. 다만 각 유닛은 한 번의 **약진**에 최대로 소모할 수 있는 스태미나 총량이 제한되어 있으며, 이를 유닛의 '**이동력**'이라고 한다. 약진이란 전투 중인 유닛이 비교적 가까운 목표지점까지 빠르게 달려 단숨에 이동하는 전술 행동으로서 하나 이상의 칸을 거쳐 이동하는 것이다. 약진은 목표지점에 다른 유닛이 없어야만 가능하다. 약진 중에 같은 세력의 유닛을 만났다면 통과하여 지나갈 수 있으나, 다른 세력의 유닛을 만났다면 그 유닛과 인접하게 되는 순간 교전이 벌어지기 때문에 그 자리에 멈춰야만 한다. 하지만, 선택된 유닛이 이미 교전 상태였다면 약진하여 교전에서 벗어날 수 있다.

당신은 게임에 버그가 있는지 테스트하기 위해서 자동으로 임의의 유닛을 선택하여 약진 명령을 내리는 봇을 만들었다. 이 봇은 수행이 불가능한 약진 명령을 내리기도 한다. 목표 지점에 다른 유닛이 있거나, 목표 지점이 이동 불가 지형이거나, 이동력의 한계로 인해 목표 지점에 도달하는 경로가 존재하지 않는 경우, 그 명령은 수행 불가능하다. 게임에 버그가 없다면 이러한 명령은 무시되어야 한다.

이제 버그가 있는지 없는지 확인할 시간이다. 봇이 내린 명령들이 시간 순으로 주어졌을 때, 모든 명령이 순차적으로 처리된 후 각 유닛이 위치한 좌표를 출력하는 프로그램을 작성하시오.

입력

첫 번째 줄에 지형의 종류 수 N, 맵의 세로 길이 H, 맵의 가로 길이 W가 공백으로 구분되어 주어진다. ($1 \le N \le 9$, $2 \le H, W \le 500$)

이어서 H개의 줄에 걸쳐 왼쪽부터 차례대로 각 칸의 지형 번호를 의미하는 W개의 정수가 공백으로 구분되어 주어지며, 각 수는 1 이상 N 이하이다.

다음 줄에 N개의 정수 $r_1, r_2, \cdots, r_N$ ($-1 \le r_i \le 4, r_i \neq 0$)이 공백으로 구분되어 주어진다. r_i가 -1이라면 i번째 지형이 너무 험준하여 진입할 수 없음을 의미하며, 이외의 경우에는 r_i는 i번째 지형의 험준도를 의미한다.

다음 줄에 유닛의 수 M이 주어진다. ($1 \le M \le H \times W / 4$)

이어서 M개의 줄에 걸쳐 1번 유닛부터 차례로 각 유닛의 이동력, 세력, 유닛이 있는 칸의 y좌표, 유닛이 있는 칸의 x좌표 정보를 의미하는 네 개의 정수 m, t, a, b가 공백으로 구분되어 주어진다. ($1 \le m \le 20$, $0 \le t \le 1$, $0 \le a < H$, $0 \le b < W$)

각 칸에는 최대 하나의 유닛이 배치되며, 진입 불가능한 지형의 칸에는 유닛이 배치되지 않는다.

다음 줄에 약진 명령의 개수 K ($1 \le K \le 10\,000$)가 주어진다.

이어서 K개의 줄에 걸쳐 약진 명령을 의미하는 세 정수 u, a, b가 공백으로 구분되어 주어진다. ($1 \le u \le M$, $0 \le a < H$, $0 \le b < W$) 이는 u번 유닛을 (a,b)로 약진시키는 명령이다.

출력

M개의 줄에 걸쳐 모든 약진 명령이 처리된 후의 유닛의 위치를 출력한다. i번 유닛이 (a_i, b_i)에 위치할 경우 i번째

줄에 두 정수 a_i, b_i를 공백으로 구분하여 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
3 5 5 1 1 3 3 2 3 3 3 1 2 1 1 1 2 1 2 2 1 1 1 1 1 1 1 3 1 3 -1 2 7 0 2 0 4 1 3 3 3 1 1 3 2 4 4 1 4 3	4 3 3 3

노트

첫 번째 약진 명령의 경우, 적대 세력 유닛을 고려하지 않는다면 $(2,0) \to (2,1) \to (2,2) \to (2,3) \to (1,3)$과 같은 경로로 이동하여 도달할 수 있다. 하지만 $(3,3)$에 위치한 적대 세력 유닛으로 인해 $(2,3)$에서 교전이 발생하므로 $(1,3)$에는 도달할 수 없고, 교전이 발생하지 않도록 우회해서 이동할 수 있는 경로가 없으므로 마찬가지로 도달할 수 없다. 따라서 이 명령은 수행 불가능하므로 무시된다.

두 번째 약진 명령은 목표 위치가 이동 불가 지형이기에 수행 불가능하므로 무시된다.

세 번째 약진 명령은 $(2,0) \to (3,0) \to (4,0) \to (4,1) \to (4,2) \to (4,3)$의 경로로 움직이면 7의 스태미나를 소모하여 이동할 수 있다. 이는 유닛의 이동력보다 크지 않은 값이므로 수행할 수 있는 명령이다.

따라서 모든 명령이 처리된 후 1번 유닛은 마지막 명령에 의해 $(4,3)$에 위치하고 2번 유닛은 초기 위치에 그대로 위치하게 된다.

문제 G. 루머

시간 제한	10 초
메모리 제한	1024 MB

당신은 루머를 믿는가?

한 유명 심리학 실험에서는 사람들에게 두 개의 줄을 보여주고, 어떤 줄이 더 긴지 말하라 했다. 사실 한 사람을 제외하고 나머지는 실험자에 의해 사전에 조작된 사람들이었다. 조작된 사람들은 사실상 더 짧은 줄을 더 길다고 말했다. 주변 모두가 같은 답변을 하자, 진짜 피실험자 또한 짧은 줄이 더 길다고 말했다. 이 실험은 사람들이 주변인의 영향을 강하게 받는다는 것을 보여주었는데, 루머도 이와 같다.

루머는 최초 유포자로부터 시작한다. 최초 유포자는 여러 명일 수 있고, 최초 유포자를 제외하고 스스로 루머를 만들어 믿는 사람은 없다.

매분 루머를 믿는 사람은 모든 주변인에게 루머를 동시에 퍼트리며, 군중 속 사람은 주변인의 절반 이상이 루머를 믿을 때 본인도 루머를 믿는다.

루머를 믿는 순간부터 다른 말은 듣지 않기 때문에, 한번 믿은 루머는 계속 믿는다.

이때, 사람들이 루머를 처음 믿기 시작하는 시간을 알아내 보자.

입력

첫째 줄에 사람의 수 N이 주어진다. $(1 \le N \le 200\,000)$ 이는 1번 사람부터 N번 사람까지 있음을 의미한다.

둘째 줄부터 N개의 줄이 주어진다. 이 중 $i(1 \le i \le N)$번째 줄에는 i번 사람의 주변인들의 번호와 입력의 마지막을 나타내는 `0`이 공백으로 구분되어 주어진다. 번호는 1 이상 N 이하의 자연수이고, 같은 줄에 중복된 번호는 없다. 자기 자신이 주변인이거나 일방적으로 주변인인 경우는 없으며, 전체 양방향 주변인 관계는 1 000 000개를 넘지 않는다.

다음 줄에는 루머를 퍼뜨리는 최초 유포자의 수 M이 주어진다. $(1 \le M \le N)$

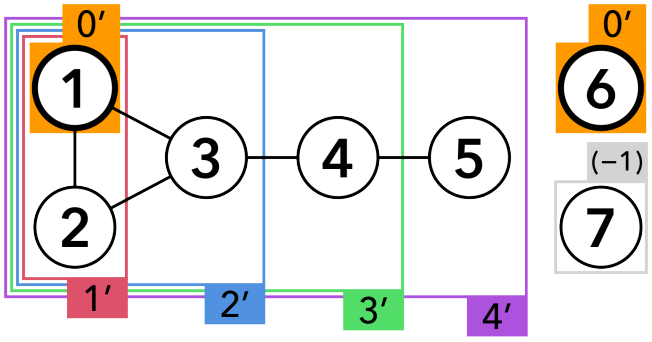
마지막 줄에는 최초 유포자의 번호가 공백으로 구분되어 주어진다. 최초 유포자의 번호는 중복되지 않는다.

출력

N개의 정수 $t_1, t_2, \cdots, t_N$을 공백 단위로 출력한다. t_i는 i번 사람이 루머를 처음 믿기 시작한 시간(분)이며, 충분히 많은 시간이 지나도 루머를 믿지 않을 경우 -1이다. 최초 유포자는 루머를 0분부터 믿기 시작했다고 생각한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
7 2 3 0 1 3 0 1 2 4 0 3 5 0 4 0 0 0 2 1 6	0 1 2 3 4 0 -1
7 2 4 0 1 3 0 2 5 0 1 5 6 0 3 4 6 7 0 4 5 7 0 5 6 0 1 6	4 4 3 3 2 0 1



예제 1

0분 최초 유포자(1, 6번 사람)가 루머를 생성한다.

1분 1번 사람은 2, 3번 사람에게 루머를 퍼뜨린다. 2번 사람은 주변인 2명 중 1명이 루머를 믿고 있어 루머를 믿게 된다. 3번 사람은 주변인 3명 중 1명만 루머를 믿기 때문에, 루머를 믿지 않는다. 6번 사람은 루머를 퍼뜨릴 주변인이 없다.

2분 1, 2번 사람은 3번 사람에게 루머를 퍼뜨린다. 인접한 3명 중 절반 이상인 2명이 루머를 믿고 있어, 3번 사람도 2분부터 믿기 시작한다.

3분 3번 사람은 4번 사람에게 루머를 퍼뜨린다. 4번 사람이 믿기 시작한다.

4분 5번 사람도 루머를 믿게 된다. 7번 사람은 이후 충분한 시간이 지나도 루머를 믿지 않는다.

문제 H. 사과나무

시간 제한	1 초
메모리 제한	1024 MB

이하는 최근 사과나무 씨앗을 구매하여 농장 뒷뜰에 일렬로 1번부터 N번까지 심었다. 이 나무들의 초기 높이는 모두 0이다.

사과나무를 무럭무럭 키우기 위해 이하는 물뿌리개 2개를 준비했다. 한 물뿌리개는 나무 하나를 1만큼 성장시키고, 다른 물뿌리개는 나무 하나를 2만큼 성장시킨다. 이 물뿌리개들은 동시에 사용해야 하며, 물뿌리개를 나무가 없는 토양에 사용할 수는 없다. 두 물뿌리개를 한 나무에 사용하여 3만큼 키울 수도 있다.

물뿌리개 관리 시스템을 전부 프로그래밍한 이하는 이제 사과나무를 키워보려고 했다. 그 순간, 갊자가 놀러와서 각 사과나무의 높이가 이런 배치가 되었으면 좋겠다고 말했다. 이제 이하는 약간 걱정이 되기 시작했는데, 갊자가 알려준 사과나무의 배치를 이 프로그램 상으로 만들어내지 못할 수도 있기 때문이다.

이하는 이제 프로그램을 다시 수정하느라 바쁘기 때문에, 두 물뿌리개를 이용해 갊자가 알려준 사과나무의 배치를 만들 수 있는지의 여부를 판단하는 과정은 여러분의 몫이 되었다.

입력

첫 번째 줄에는 자연수 N이 주어진다. ($1 \le N \le 100\,000$) 이는 이하가 뒷뜰에 심은 사과나무의 개수를 뜻한다.

두 번째 줄에는 N개의 정수 $h_1, h_2, \cdots, h_N$이 공백으로 구분되어 주어진다. ($0 \le h_i \le 10\,000$)
h_i는 갊자가 바라는 i번째 나무의 높이이다.

출력

첫 번째 줄에 모든 나무가 갊자가 바라는 높이가 되도록 물뿌리개를 통해 만들 수 있으면 "YES"를, 아니면 "NO"를 따옴표를 제외하고 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
1 0	YES
2 4 3	NO
3 10000 1000 100	YES
5 1 3 1 3 1	NO

이 페이지는 공백입니다

문제 I. 인버스 ㄷㄷㄷㅈ

시간 제한	2 초
메모리 제한	1024 MB

UCPC 출제진들은 D번: ㄷㄷㄷㅈ 문제의 데이터를 만들던 중, 정점이 많은 DUDUDUNGA-트리를 만드는 것이 어렵다는 것을 알게 되었다. N이 주어졌을 때, 정점이 N개인 DUDUDUNGA-트리를 출력하는 프로그램을 만들어보자.

입력

첫 번째 줄에 트리의 정점 수 N이 주어진다. $(6 \leq N \leq 300\,000)$

출력

$N-1$개의 줄에 간선의 양 끝 점을 공백으로 구분해서 출력한다. 정점의 번호는 1 이상 N 이하의 정수여야 한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
6	1 2 2 3 3 4 4 5 4 6

노트

DUDUDUNGA-트리의 정의는 D번: ㄷㄷㄷㅈ 문제를 참고하자. 입력으로 주어지는 N에 대해 언제나 정점의 개수가 N개인 DUDUDUNGA-트리가 존재한다.

이 페이지는 공백입니다

문제 J. 역학 조사

시간 제한	3 초
메모리 제한	1024 MB

2020년, 신종 전염병이 유행하여 UCPC국 질병관리본부에서 역학 조사를 하고 있다. UCPC국의 인구는 총 N명이며 각각 $1, 2, \cdots, N$번의 주민번호가 붙어있다.

질병관리본부는 지금까지 M개의 모임이 있었다는 사실을 파악했다. 각 모임은 k명이 참여했고 해당 모임은 주민번호가 $a_1, a_2, \cdots, a_k$인 사람들이 참여했다.

전염병은 밀접하고 밀폐된 공간에서만 전염되기 때문에 반드시 모임 안에서만 전염된다. 전염병이 전파되는 규칙은 다음과 같다.

- 모임에 참여한 사람들 중 한 명 이상의 사람이 전염병에 감염되어 있었다면 모임에 참여한 모든 사람들이 전염병에 감염된다.
- 모임에 전염병에 감염된 사람이 없다면 아무 일도 일어나지 않는다.

질병관리본부는 확보한 자료를 가지고 초기 감염자들을 예측하려고 한다. 모임의 정보 및 M개의 모임이 끝나고 나서 전염병에 감염된 사람의 정보가 주어지면 첫 번째 모임을 하기 전에 감염되어 있던 사람을 역추적하는 프로그램을 작성하여라. 위 규칙 이외의 경로로 전염병이 전파되거나 전염병이 치료되는 경우는 없다고 간주한다.

입력

첫 번째 줄에 사람의 수 N, 모임의 수 M $(2 \le N \le 100\,000, 1 \le M \le 100\,000)$이 주어진다.

두 번째 줄부터 M개의 줄에는 모임의 정보가 시간 순으로 주어진다. 각 줄에는 각 모임에 참여하는 사람의 수 k $(2 \le k \le N)$와 모임에 참여한 사람의 주민번호 a_i $(1 \le a_i \le N, a_i \ne a_j)$가 주어진다. 여러 모임이 동시에 진행되는 경우는 없다.

마지막 줄에는 N명의 사람들에 대한 감염 정보가 주어진다. 마지막 모임이 끝나고 주민번호가 i인 사람이 전염병에 감염되었다면 `1`을, 그렇지 않다면 `0`이 주어진다. 감염된 사람이 없을 수 있음에 유의하여라.

k들의 합은 1 000 000을 넘지 않는다.

출력

만약 모임을 하기 전에 감염된 사람을 역추적할 수 없다면 `NO`를 출력한다.

그렇지 않다면 첫 번째 줄에 `YES`를 출력하고 두 번째 줄에 감염 정보를 의미하는 정수 N개를 공백으로 구분하여 출력한다. 이 중 i번째 수는 주민번호가 i인 사람이 첫 번째 모임이 시작되기 전에 전염병에 감염되어 있었으면 `1`이며, 아니면 `0`이다. 가능한 감염 상태가 여러 개이면, 그중 아무 것이나 출력하면 된다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
7 3 3 1 2 3 3 3 4 5 3 5 6 7 0 0 1 1 1 1 1	YES 0 0 0 1 1 1 1
7 3 3 1 2 3 3 3 4 5 3 5 6 7 1 0 1 0 1 0 1	NO

이 페이지는 공백입니다