

snpcc

2021

2021 서울대학교 프로그래밍 경진대회

Division 1

주최 및 주관



후원

NAVER



서울대학교 컴퓨터공학부

Seoul National University
Dept. of Computer Science and Engineering



참가자를 위한 도움말

주의 사항

- 대회 시간은 12:30부터 17:30까지입니다. 대회가 진행되는 동안 인터넷 검색 등을 하실 수 있습니다. 단, 인터넷에 있는 코드를 그대로 복사-붙여넣기하시는 경우, 유사도 검사 시 확인을 위해 **코드의 출처(인터넷 URL 등)**를 주석으로 적어주시기 바랍니다. 각 언어의 레퍼런스 사이트는 다음과 같습니다.
 - C/C++ reference: <https://en.cppreference.com/w/>
 - Python 3 reference: <https://docs.python.org/3.9/>
 - Java documentation: <https://docs.oracle.com/javase/8/docs/api/>
 - Kotlin documentation: <https://kotlinlang.org/docs/home.html>
- 대회는 Baekjoon Online Judge(<https://www.acmicpc.net/>) 플랫폼을 이용하여 진행됩니다. 별도로 제공되는 계정 정보를 이용하여 로그인하신 뒤 코드 제출 및 결과 확인 등을 하실 수 있습니다.
- 제출하실 답안 코드는 **C11, C++17, C++20, Java 8, Java 8 (OpenJDK), Python 3.9.5, PyPy3, Kotlin (JVM) 으로만** 작성하셔야 합니다. 단, 출제진은 C++20을 제외한 다른 언어로 주어진 시간 제한과 메모리 제한을 지키며 올바른 답을 내는 코드를 작성할 수 있다는 보장을 하지 않습니다.
- 모든 입력은 **표준 입력**으로 주어지며, 모든 출력은 **표준 출력**으로 합니다.
- 테스트 케이스가 존재하는 문제의 경우, 테스트 케이스에 대한 출력을 모아서 하실 필요 없이, 각 테스트 케이스를 처리할 때마다 출력해도 괜찮습니다.
- **리턴 코드와 표준 오류(standard error, stderr) 스트림 출력**에 주의하십시오. 프로세스가 0이 아닌 리턴 코드를 되돌리는 경우나 **표준 오류 스트림에 출력**하는 경우 “**런타임 에러**”를 받게 됩니다.
- 문제에 대한 질의 사항은 대회 페이지의 질문 기능을 사용해 주시기 바랍니다. 이 때 대답해 드리기 어려운 질문에 대해서는 “**답변을 드릴 수 없습니다**”로 대답될 수 있으므로 유의하십시오.
- 이 문제지는 참고용입니다. 문제지에 적힌 문제와 플랫폼에 올라온 문제가 다를 경우, 플랫폼의 문제가 우선합니다. 특히 이 문제지를 흑백으로 프린트하신 경우 색이 들어간 이미지는 플랫폼에서 보시기 바랍니다.

채점 결과에 대하여

맞았습니다!! 제출하신 답안이 모든 테스트 데이터를 정해진 시간과 메모리 제한 안에 통과하여 정답으로 인정되었음을 의미합니다.

틀렸습니다 제출하신 답안 프로그램이 테스트 데이터에 대해 생성한 출력이 출제자의 정답과 일치하지 않음을 의미합니다.

컴파일 에러 제출하신 답안 프로그램을 컴파일하는 도중 오류가 발생하였음을 의미합니다.

런타임 에러 제출하신 답안 프로그램을 실행하는 도중 프로세스가 비정상적인 행동을 하였음을 의미합니다.

시간 초과 제출하신 답안 프로그램이 정해진 시간 안에 종료되지 않았음을 의미합니다.

메모리 초과 제출하신 답안 프로그램이 정해진 메모리 제한보다 많은 메모리를 사용하였음을 의미합니다.

출력 형식이 잘못되었습니다 답을 올바르게 구했으나, 답안 프로그램의 출력 형식이 문제에 나와 있는 출력 형식과 다름을 의미합니다. 줄 뒤에 의미 없는 공백을 두 칸 이상 출력하거나 의미 없는 빈 줄을 출력할 경우 이 결과를 받을 수 있습니다.

출력 초과 답안 프로그램이 정답에 비해 너무 많은 출력을 했음을 의미합니다. 이 결과는 틀렸습니다와 같은 의미를 갖습니다.

만약 여러 가지의 원인으로 인해 “맞았습니다!!”가 아닌 다른 결과를 얻으셨다면, 그중 어떤 것도 결과가 될 수 있습니다. 예를 들어 답도 잘못되었고 비정상적인 동작도 수행하는 코드를 제출하신 경우 대부분 “런타임 에러”를 받으시게 되지만, 경우에 따라서 “틀렸습니다”를 받을 수도 있습니다.

문제 목록

문제지에 있는 문제가 총 11 문제가 맞는지 확인하시기 바랍니다.

- A 트리 조각하기
- B 별 보는 교준이
- C 큰 수 뒤집기
- D 여우 국수
- E 자연수 색칠하기
- F 고슴도치 그래프
- G 데칼코마니 트리
- H RMQ
- I 두 트리
- J 문자열 X
- K 누텔라 트리 (Hard)

모든 문제의 메모리 제한은 1GB로 동일합니다.

문제 A. 트리 조각하기

시간 제한 2 초
메모리 제한 1024 MB

타고난 트리 조각가 온조는 오늘도 완벽한 작품을 만들기 위해서 트리 T 를 준비하였다. 온조는 뛰어난 예술적 직관을 통해 조각할 작품을 머릿속으로 구상하였고, 먼저 조각상의 전체적인 틀을 잡기 위해서 T 에서 제거할 정점들을 결정하였다.

그런데 T 의 정점들은 매우 단단하기 때문에 일반적인 도구로는 제거할 수가 없고, 폭탄을 사용하여 제거해야 한다. 그래서 온조는 T 의 몇몇 정점에 폭탄을 설치한 뒤 한 번에 폭발시키는 방법을 사용하려고 한다. 모든 폭탄의 세기는 양의 정수인 p 로 같으며, 모든 폭탄이 폭발하고 난 후 폭탄이 설치된 정점과의 거리가 p 미만인 정점들은 제거된다. 이 과정에서 제거해야 할 정점이 제거되지 않거나 제거하지 않아야 할 정점이 제거되면 작품이 망가지기 때문에, 온조는 이런 일이 일어나지 않도록 폭탄을 설치할 것이다.

온조는 폭발 과정도 작품의 한 부분이기 때문에 폭탄의 세기 p 가 크면 클수록 작품의 예술적 가치가 높아진다고 생각한다. 트리 T 와 제거해야 할 정점들이 주어지면 폭탄의 세기 p 로 가능한 값 중 최댓값을 구해서 온조를 도와주자. 모든 정점을 제거해야 하는 경우나 모든 정점을 제거하지 않아야 하는 경우는 주어지지 않는다.

입력

첫째 줄에 트리 T 를 구성하는 정점의 개수 N 이 주어진다. ($2 \leq N \leq 200000$)

둘째 줄에 N 개의 수 C_i 가 공백을 사이에 두고 주어진다. C_i 는 0 또는 1이다. $C_i = 1$ 인 경우 i 번 정점을 제거해야 한다는 의미이며, $C_i = 0$ 인 경우 i 번 정점을 제거하지 않아야 한다는 의미이다.

셋째 줄부터 $(N-1)$ 개 줄에 걸쳐 간선 정보가 주어지며, 각 줄에는 하나의 간선이 잇는 두 정점의 번호 u, v 가 공백을 사이에 두고 주어진다. ($1 \leq u \leq N, 1 \leq v \leq N, u \neq v$)

출력

첫째 줄에 폭탄의 세기 p 로 가능한 값 중 최댓값을 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
12 0 0 0 1 1 1 0 1 0 0 0 0 11 6 4 5 6 5 7 4 8 6 2 3 3 12 11 10 1 10 1 9 3 11	2

문제 B. 별 보는 교준이

시간 제한	1.5 초
메모리 제한	1024 MB

교준이는 이차원 좌표평면의 밤하늘을 바라보고 있다. 지금 하늘에는 서로 다른 위치에 N 개의 별이 반짝이고 있고, i 번 별의 위치는 (X_i, Y_i) 이다.

교준이는 다음 조건을 모두 만족하도록 N 개의 별을 두 개의 별자리, A 별자리와 B 별자리로 구분하고자 한다. 교준이가 사는 세상에서 별은 크기가 없는 고정된 점임에 유의하라.

- 각 별은 A 별자리와 B 별자리 중에서 정확히 하나의 별자리에 속한다.
- A 별자리 혹은 B 별자리에 속하는 별은 모두 교준이가 바라보고 있는 N 개의 별 중 하나이다.
- A 별자리와 B 별자리는 각각 적어도 하나의 별을 가진다.
- A 별자리에 속한 별과 B 별자리에 속한 별을 서로 분리하는, N 개의 별 중 어느 하나도 지나지 않는 밤하늘 위의 직선이 존재한다.

위의 조건을 모두 만족하도록 N 개의 별을 A 별자리와 B 별자리로 구분하는 경우의 수를 구하라.

입력

첫째 줄에 정수 N 이 주어진다. ($2 \leq N \leq 1500$)

둘째 줄부터 N 개의 줄에 걸쳐, N 개의 별에 대한 정보가 주어진다. $(i+1)$ 번째 줄에는 두 정수 X_i, Y_i 가 주어진다. ($1 \leq i \leq N, |X_i| \leq 10^8, |Y_i| \leq 10^8$)

출력

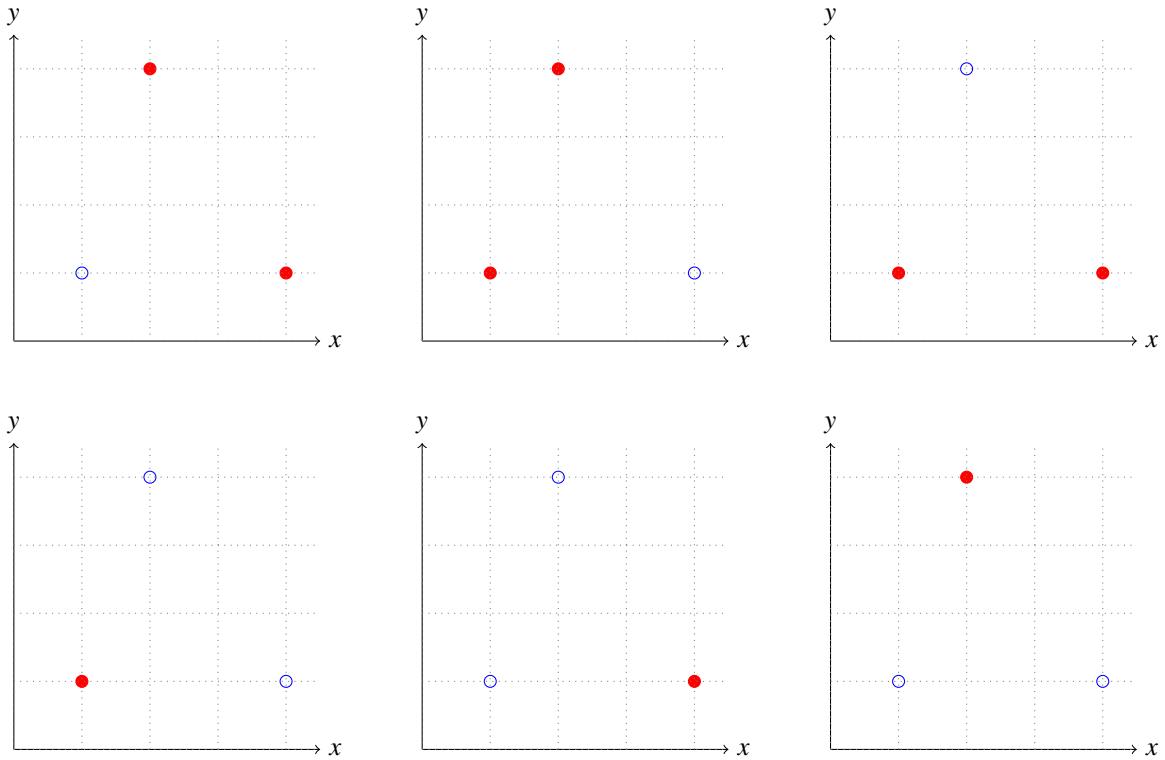
첫째 줄에 경우의 수를 $10^9 + 7$ 로 나눈 나머지를 출력한다.

입출력 예시

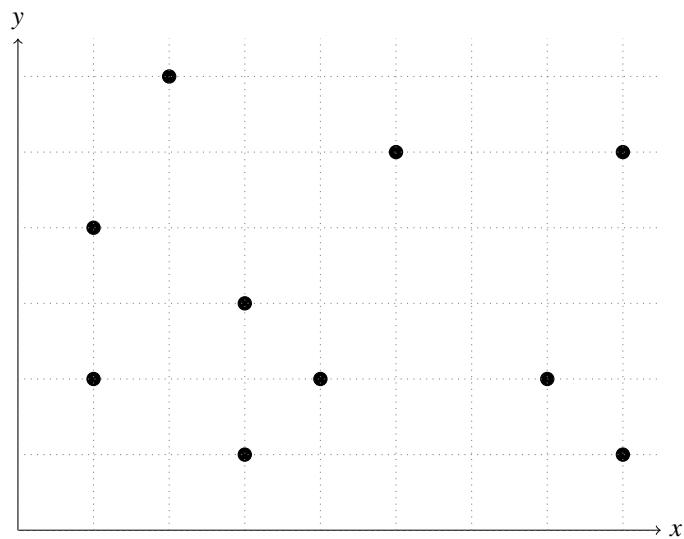
표준 입력(stdin)	표준 출력(stdout)
3 1 1 4 1 2 4	6
3 -1 -1 0 0 1 1	4
10 3 3 2 6 8 5 1 4 8 1 7 2 5 5 1 2 3 1 4 2	88

노트

첫째 예제의 경우, 별을 구분하는 방법은 총 6가지다. 속이 찬 빨간 원과 속이 빈 파란 원은 각각 A 별자리와 B 별자리에 속하는 별을 의미한다.



마지막 예제를 그림으로 나타내면 아래와 같다.



문제 C. 큰 수 뒤집기

시간 제한 3 초
메모리 제한 1024 MB

문자열 S 와 정수 X 가 있다. 처음에 S 는 빈 문자열이며 $X = 0$ 이다. 이때, 다음 쿼리를 수행해야 한다.

- 1부터 9까지의 숫자 중 하나가 주어지면, S 의 맨 뒤에 그 숫자를 덧붙인다. 그리고 X 에 S 를 10진법 정수로 읽었을 때의 값을 더한다.
- 하이픈(-)이 주어지면, S 를 뒤집는다.

모든 쿼리를 처리한 뒤 X 의 값을 구하는 프로그램을 작성하자.

입력

첫째 줄에 쿼리 정보를 담고 있는 길이 1 이상 2000000 이하의 문자열 T 가 주어진다. 각 문자는 1부터 9까지의 숫자 중 하나 또는 -이며, 첫 번째 문자는 -이 아니다.

출력

쿼리를 순서대로 모두 처리한 뒤 X 의 값을 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
1234-567-89-123	1093443373651
12345-12345-12345-12345-12345	6035727023470496707633735

노트

정답이 매우 클 수 있음에 유의하자.

문제 D. 여우 국수

시간 제한 0.5 초
메모리 제한 1024 MB

관악산 깊은 곳에는 여우들과 너구리들이 살고 있는 마을이 있다. 이 동물들은 변신에 능하기 때문에, 자주 사람으로 변신해서 근처 대학교에 놀러 가곤 한다.

어느 날, 마을에서 학교로 내려가는 길에 장인 여우가 국수 가게를 차렸다. 이 가게의 대표 메뉴는 유부가 들어간 **여우 국수**와 튀김 부스러기가 들어간 **너구리 국수**이다. 이 가게는 곧 마을의 여우들과 너구리들에게 큰 인기를 끌게 되었다. 물론 여우들은 가게에 방문하면 반드시 여우 국수를 주문하고, 너구리들은 반드시 너구리 국수를 주문한다.

장인 여우는 오랜 경험으로 하루에 몇 마리의 여우와 너구리가 올지 정확히 예측할 수 있다. 장인 여우는 내친김에 이 손님들의 순서까지 맞추는 연습도 하고 있지만, 아직은 많이 서투르다. 그래도 장인 여우는 매일 아침 손님들의 순서를 예측하고, 반드시 그 순서에 따라서 국수를 조리한다. 또한 장인 여우는 국수가 만들어진 시점을 구분하기 위해 한 그릇을 조리할 때마다 1번부터 차례대로 번호를 매긴다.

만약 예측한 순서가 틀리면, 국수를 조리되는 대로 손님에게 내어 주면 어떤 손님들은 주문과 다른 국수를 받게 된다. 이를 방지하기 위해, 장인 여우는 음식을 올려놓으면 맛과 온도를 오랫동안 유지해 주는 마법 진열대를 마련했다. 장인 여우는 국수를 조리하면 즉시 진열대 위에 올려놓고, 손님에게 국수를 내놓을 때는 항상 진열대 위에 있는 국수들 중 가장 나중에 조리된 것, 즉 번호가 가장 큰 것을 꺼내서 내놓는다. 매 시점에 새로운 국수를 조리할지, 아니면 손님에게 국수를 내놓을지는 장인 여우가 마음대로 선택할 수 있다.

이 국수 가게에는 항상 첫 번째로 찾아오는 단골 손님이 있다. 장인 여우는 이 손님에게 소소한 **행운의 번호** 이벤트를 제공하기로 했다. 첫 번째 손님이 어떤 번호 n 을 부르면, 장인 여우는 n 번 국수를 이 손님에게 내어야 한다. 이때, n 번 국수가 첫 번째 손님이 주문한 메뉴가 아니거나, 장인 여우가 아무리 노력하더라도 그 뒤로 찾아온 모든 손님의 주문을 만족시킬 수 없다면 **팍** 판정을 받는다. 그렇지 않을 경우는 **당첨** 판정을 받고, 첫 번째 손님은 열렬한 축하를 선물로 받는다.

장인 여우가 예측한 손님들의 순서와 실제로 찾아온 손님들의 순서가 주어질 때, 가능한 당첨 번호들을 모두 구해보자.

입력

첫째 줄에 하루 동안 찾아온 손님들의 총 마릿수를 나타내는 정수 $N(1 \leq N \leq 5000000)$ 이 주어진다.

둘째 줄에 장인 여우가 예측한 손님들의 순서를 나타내는 N 글자의 문자열이 주어진다. 이 문자열은 알파벳 **Y**와 **N**으로만 구성되어 있으며, **Y**는 여우, **N**은 너구리를 의미한다.

셋째 줄에 실제로 찾아온 손님들의 순서를 나타내는 N 글자의 문자열이 두 번째 줄과 같은 형식으로 주어진다.

두 문자열에 들어 있는 알파벳 **Y**의 개수는 서로 같으며, 자연히 알파벳 **N**의 개수 역시 서로 같다.

출력

첫째 줄에 첫 번째 손님이 부를 수 있는 당첨 번호의 총합을 출력한다. 만약 어떤 번호를 불러도 팍 판정을 받게 된다면 **0**을 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
3 YNN NNY	5
4 NYNY YNNY	2

노트

첫 번째 예시에서 행운의 번호로 1을 부를 경우, 첫 번째 손님은 너구리인데 여우 국수를 받게 되므로 짱 판정을 받는다.

2를 부를 경우는 다음과 같은 방법으로 장인 여우가 모든 주문을 처리할 수 있으므로 당첨 판정을 받는다.



마찬가지로 3도 당첨 번호임을 쉽게 알 수 있으며, 답으로 $2+3=5$ 를 출력한다.

두 번째 예시에서 행운의 번호로 4를 부를 경우 메뉴는 올바르지만, 이후 진열대 위에 남은 국수들을 너구리 → 여우 → 너구리 순으로 내놓게 되므로 남은 손님들의 주문을 만족시킬 수 없다.

문제 E. 자연수 색칠하기

시간 제한 1 초
메모리 제한 1024 MB

1부터 N 까지의 자연수를 색칠한다. 단, 서로소인 서로 다른 두 자연수는 다른 색으로 칠해야 한다. 최소한의 색을 써서 모든 자연수를 칠하는 방법을 찾는 프로그램을 작성하자.

입력

첫째 줄에 자연수 N 이 주어진다. ($1 \leq N \leq 500000$)

출력

첫째 줄에 사용한 색의 수 K 를 출력한다.

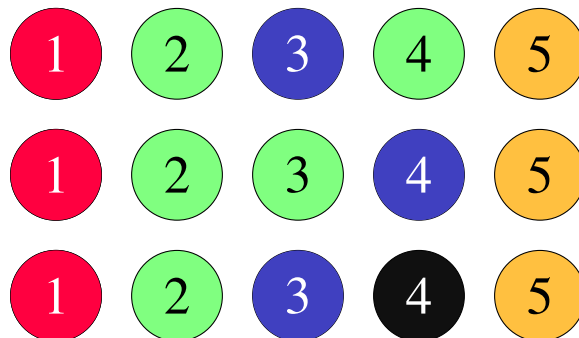
둘째 줄에 N 개의 수를 공백을 사이에 두고 출력한다. i 번째 수는 자연수 i 의 색이다. 색은 1 이상 K 이하의 정수로 나타낸다.

입출력 예시

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

노트

다음은 $N = 5$ 일 때의 올바른 색칠과 올바르지 않은 색칠의 예시다.



첫 번째 색칠은 **올바른 색칠**이다. 2와 4는 같은 색으로 색칠했지만, 서로소가 아니므로 문제의 조건에 위배되지 않는다. 또한, 5까지의 자연수를 색칠하기 위해 최소 4개의 색이 필요함을 증명할 수 있다.

두 번째 색칠은 서로소인 2와 3이 같은 색으로 칠해졌기 때문에 **올바르지 않은 색칠**이다.

세 번째 색칠은 최소한의 색깔로 칠하지 않았기 때문에 **올바르지 않은 색칠**이다.

문제 F. 고슴도치 그래프

시간 제한 2.5 초
메모리 제한 1024 MB

이 문제는 인터랙티브 문제입니다.

고슴도치 귀엽지 않나요?



그림 1: 귀여운 고슴도치!

키파는 “성계 그래프”도 있는데 이렇게 귀여운 고슴도치에 대응하는 그래프가 없는 것은 아무래도 말이 안 된다고 생각해서, **고슴도치 그래프**를 만들기로 했습니다.

고슴도치 그래프는 다음과 같은 두 가지 성질을 만족하는 방향 없는 연결 그래프입니다. 이 그래프에는 self-loop(한 정점 v 와 같은 정점인 v 를 연결하는 간선)와 multiedge(서로 다른 두 정점 v 와 w 에 대해 v 와 w 를 연결하는 서로 다른 간선 e_1 과 e_2)가 없습니다.

- 사이클이 정확히 하나 존재합니다. 이 사이클을 고슴도치의 **몸통**이라고 부르겠습니다. 몸통의 **크기**는 사이클에 속한 정점의 개수입니다. 사이클이 항상 존재하기 때문에 **몸통의 크기는 항상 3 이상**이라는 점에 유의하세요.
- 사이클에 속하지 않은 정점의 차수가 1입니다.

키파는 고슴도치 그래프를 열심히 키운 결과 각 그래프가 $V = 10^6$ 개의 정점으로 이루어진 고슴도치 그래프를 총 N 개 가지게 되었습니다. 키파는 각 고슴도치 그래프를 다음과 같은 방법으로 어루만져 주기로 했습니다.

1. 각 노드에 붙어 있는 간선의 방향을 한 방향으로 정하되, 모든 노드가 나가는 방향의 간선을 **정확히 하나**

가지도록 정합니다. 임의의 고슴도치 그래프에 대해, 간선의 방향을 위 조건을 만족하도록 정하는 것이 항상 가능함을 증명할 수 있습니다.

2. 다음을 최대 $M = 1204$ 번 반복합니다.

- (a) 정점 v 와 $\lfloor 10^{12.4} \rfloor = 2511886431509$ 이하의 아무 양의 정수 x 를 고릅니다.
- (b) v 에서 화살표를 x 번 따라가며 고슴도치 그래프를 쓰다듬습니다. 모든 노드가 나가는 방향의 간선이 하나 있기 때문에 도착 노드는 하나로 정해지며, 이 도착 노드가 어디인지 알 수 있습니다.

과정 1은 키파가 이미 임의로 방향을 정했습니다.

키파는 고슴도치 그래프를 어루만져 주는 것뿐만 아니라 각 고슴도치 그래프의 몸통의 크기를 알고 싶어했습니다. 하지만 키파는 이걸 생각하기에는 머리가 너무 아팠기 때문에, 과정 2를 여러분에게 맡기기로 했습니다.

여러분은 각 고슴도치 그래프에 대해 과정 2를 시행한 후 얻은 정보로 각 고슴도치의 몸통의 크기를 알아내야 합니다.

입력

첫째 줄에 10 이하의 양의 정수 N 이 주어집니다. 그리고 나면 인터랙션이 시작됩니다.

상호 작용

여러분은 채점 시스템에게 다음과 같은 질의를 할 수 있습니다.

- **? v x**: 정점 v 에서 화살표를 x 번 따라가서 도착한 노드 번호를 돌려줍니다. $1 \leq v \leq V$, $1 \leq x \leq \lfloor 10^{12.4} \rfloor = 2511886431509$ 를 만족해야 합니다. 이 질의는 그래프 당 최대 M 번 할 수 있으며, 결과는 도착 정점 번호를 나타내는 1 이상 V 이하의 정수입니다.
- **! s**: 현재 어루만지고 있는 그래프의 몸통의 크기 s 를 채점 프로그램에게 알려주는 질의입니다. 그래프 당 한 번 질의할 수 있으며, 결과는 몸통의 크기가 맞았음을 의미하는 1이거나 몸통의 크기가 틀렸음을 의미하는 -1 입니다. 크기가 맞은 경우 다음 그래프로 넘어갑니다.

질의를 출력한 후에는 표준 출력 버퍼를 flush해 주어야 합니다. 언어별로 표준 출력 버퍼를 flush하는 방법은 다음과 같습니다.

- C: `fflush(stdout)`
- C++: `std::cout << std::flush`
- Java: `System.out.flush()`
- Python: `sys.stdout.flush()`
- Kotlin: `System.out.flush()`

어떤 질의에서든지 음수를 응답으로 받았을 경우 프로그램을 즉시 정상 종료해야 하며, 이 경우 **틀렸습니다**를 받습니다. 프로그램을 즉시 종료하지 않을 경우 다른 결과를 받을 수 있음에 주의하세요.

N 개의 **!** 질의에서 모두 1을 받은 경우 프로그램을 종료하면 그 테스트 케이스에 대해서는 맞은 것으로 처리됩니다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
1	? 1 2
3	? 2 5
7	? 10 11
10	! 11
1	

문제 G. 데칼코마니 트리

시간 제한 4 초
메모리 제한 1024 MB

아래의 아름다운 나비를 보아라.

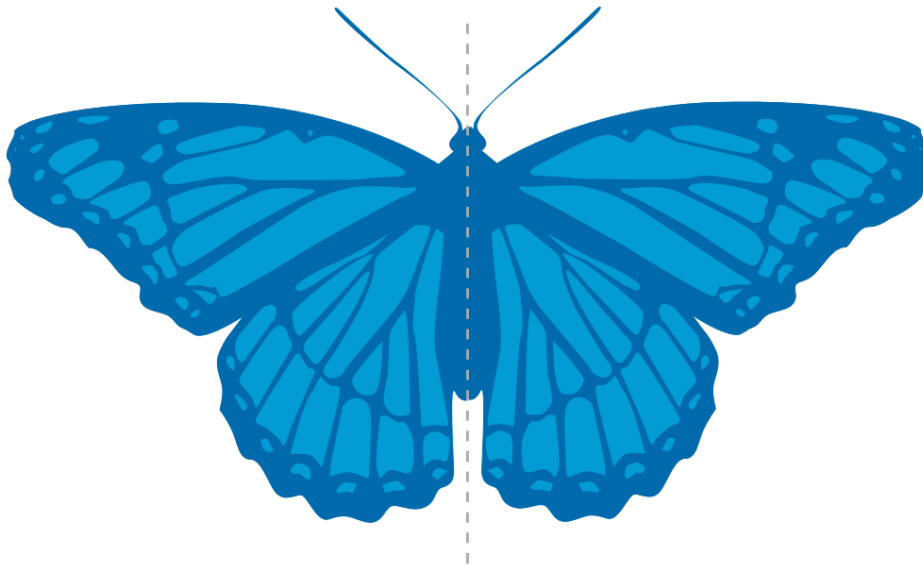


그림 1: 아름다운 나비

위 그림은 회색 점선으로 표시된 직선에 대하여 대칭을 이루고 있다. 이렇듯 선대칭을 이루는 그림을 “데칼코마니”라고 부른다.

N 개의 정점으로 이루어진 트리에 대하여 **트리 그림**을 정의하자.

- 트리 그림은 원 N 개와 선분 $(N - 1)$ 개로 이루어져 있다.
- 하나의 원은 트리에서 하나의 정점을 의미한다. 원과 정점은 서로 일대일 대응된다.
- 하나의 선분은 트리에서 하나의 간선을 의미한다.
 - 이 선분의 양 끝 점 각각은 어떤 원의 둘레 위에 있다. 끝 점이 두 개이므로 이 조건을 만족하는 원은 두 개인데, 이 두 원을 선분의 **양 끝 원**이라고 부르자.
 - 이 선분의 양 끝 점을 지나는 **직선**은 양 끝 원의 중심을 지난다.
 - 이 선분의 양 끝 원은 간선의 양 끝의 두 정점과 서로 대응된다.
- 원은 다른 원과 점을 공유해서는 안 되며, 선분은 다른 선분이나 원과 점을 공유하지 않아야 한다. 단, 선분의 조건에 의해 어떤 선분이 양 끝 원과 양 끝 점을 공유하는 것은 허용된다.
- 모든 원의 반지름은 동일하며, 선분의 두께는 무시 가능할 정도로 아주 얇다.

다음은 트리 그림의 예시를 나타낸 것이다.

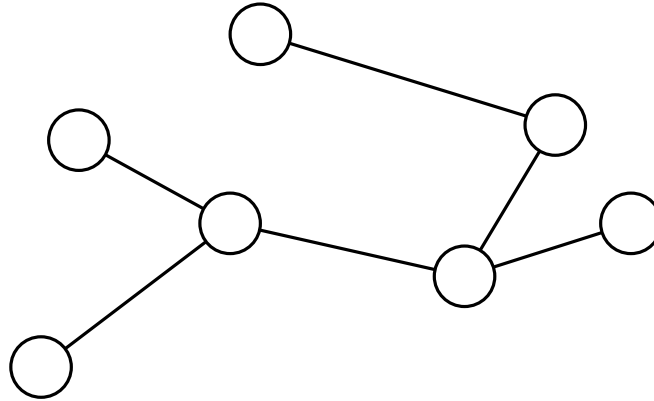


그림 2: 트리 그림

어떤 트리의 트리 그림 중 데칼코마니인 것이 존재한다면 이러한 트리를 **데칼코마니 트리**라고 부르자. 정점 N 개로 이루어진 트리가 주어질 때, 이 트리가 데칼코마니 트리인지 판별하라.

입력

첫째 줄에 정수 N 이 주어진다. ($1 \leq N \leq 10^6$)

둘째 줄부터 $(N-1)$ 개의 줄에 걸쳐 트리의 간선의 정보가 주어진다. $(i+1)$ 번째 줄에는 i 번째 간선의 정보를 나타내는 두 정수 A_i, B_i 가 주어진다. 이는 i 번째 간선이 A_i 번 정점과 B_i 번 정점을 서로 연결한다는 의미이다. ($1 \leq i \leq N-1$, $1 \leq A_i \leq N$, $1 \leq B_i \leq N$, $A_i \neq B_i$)

출력

만약 주어진 트리가 데칼코마니 트리가 아니라면 첫째 줄에 “NO”를 출력한다.

만일 주어진 트리가 데칼코마니 트리라면 첫째 줄에 “YES”를 출력한다. 이후 둘째 줄에 N 개의 정수 $C_1, \dots, C_N$ 을 공백을 사이에 두고 출력한다. 이는 데칼코마니 트리 그림에서 i 번 정점의 원과 C_i 번 정점의 원이 서로 선대칭을 이룸을 의미한다. ($1 \leq i \leq N$)

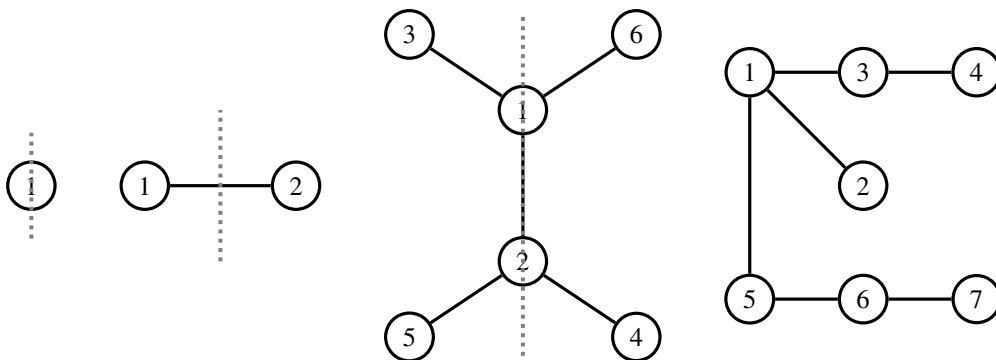
트리 그림 중 데칼코마니인 것이 여러 개 존재한다면 그중 아무거나 출력해도 정답으로 인정된다.

입출력 예시

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

노트

네 개의 예제를 그림으로 나타내면 아래와 같다.



문제 H. RMQ

시간 제한 3 초
메모리 제한 1024 MB

1 이상 N 이하의 정수가 한 번씩 등장하는 길이 N 의 수열 A 가 주어진다.

수열 A 에 대한 RMQ란 보통 Range Minimum Query 또는 Range Maximum Query를 뜻하는 말로, $1 \leq l \leq r \leq N$ 인 l 과 r 이 주어졌을 때 $\min(A_l, \dots, A_r)$ 또는 $\max(A_l, \dots, A_r)$ 를 구하는 쿼리를 의미한다.

어느 쿼리도 포기하지 못했던 종영이는 새로운 RMQ를 만들기로 했다. 종영이가 새로 만든 RMQ(Range Mueonga Query)는 $\text{RMQ}(l, r) = \min(A_l, \dots, A_r) \times \max(A_l, \dots, A_r)$ 를 의미한다.

모든 쿼리 값들을 기억하기 위해, N 행 N 열의 2차원 배열 B 에 대해 i 행 j 열의 위치에 해당하는 값 $B_{i,j}$ 에 $i \leq j$ 라면 $\text{RMQ}(i, j)$ 를, $i > j$ 라면 0을 써 놓았다.

쿼리 중독증인 종영이는 이제 B 에서 2차원 쿼리를 날리고 싶어졌다. 종영이를 위해 $1 \leq l \leq r \leq N$ 이고 $1 \leq s \leq e \leq N$ 인 네 정수 l, r, s, e 가 주어질 때마다 $l \leq i \leq r$ 이고 $s \leq j \leq e$ 인 모든 i, j 에 대해 $B_{i,j}$ 의 합을 구해주자.

입력

첫째 줄에 N 과 Q 가 공백으로 구분되어 주어진다. ($1 \leq N \leq 150000, 1 \leq Q \leq 150000$)

둘째 줄에 $A_1, \dots, A_N$ 이 공백으로 구분되어 주어진다. ($1 \leq A_i \leq N$)

셋째 줄부터 Q 개의 줄에 걸쳐 쿼리들이 주어진다. 각 쿼리에서는 l, r, s, e 가 공백으로 구분되어 주어진다. ($1 \leq l \leq r \leq N, 1 \leq s \leq e \leq N$)

출력

각 줄에 쿼리의 답을 $10^9 + 7$ 로 나눈 나머지를 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
10 10	40
7 6 1 10 5 3 2 4 8 9	24
1 2 7 8	82
7 9 8 8	0
2 10 8 8	140
5 10 1 4	278
7 9 2 9	381
1 7 8 10	260
3 8 5 9	370
3 10 2 6	201
1 4 4 8	
1 3 1 5	

문제 I. 두 트리

시간 제한	1.5 초
메모리 제한	1024 MB

N 개의 정점과 서로 다른 $2(N-1)$ 개의 간선으로 이루어진 그래프가 주어진다. 각 정점은 1번부터 N 번까지, 각 간선은 1번부터 $2(N-1)$ 번까지 번호가 부여되어 있다. i 번 간선은 A_i 번 정점과 B_i 번 정점을 서로 연결한다. ($1 \leq i \leq 2(N-1)$)
각 간선을 빨강 혹은 파랑으로 칠하자. 빨간 간선으로 이루어진 그래프와 파란 간선으로 이루어진 그래프가 각각 트리가 되도록 간선을 칠할 수 있는지 판별하라.

입력

첫째 줄에 정수 N 이 주어진다. ($4 \leq N \leq 3000$)

둘째 줄부터 $2(N-1)$ 개의 줄에 걸쳐 그래프의 간선이 주어진다. $(i+1)$ 번째 줄에는 두 정수 A_i, B_i 가 주어진다. ($1 \leq i \leq 2(N-1), 1 \leq A_i \leq N, 1 \leq B_i \leq N, A_i \neq B_i$)

각 간선이 연결하는 정점쌍은 모두 다르다. ($1 \leq i < j \leq 2(N-1), \{A_i, B_i\} \neq \{A_j, B_j\}$)

출력

만일 불가능하다면 첫째 줄에 “NO”를 출력하라.

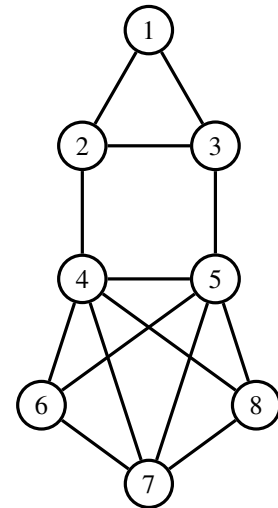
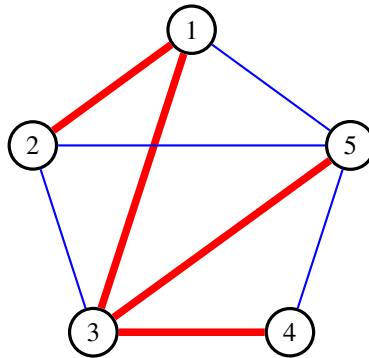
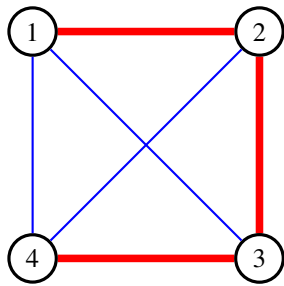
만약 가능하다면 첫째 줄에 “YES”를 출력하라. 이후 둘째 줄에 R과 B로 이루어진 길이 $2(N-1)$ 의 문자열을 출력하라. 이 문자열의 i 번째 문자가 R이라는 것은 i 번 간선을 빨강으로 칠해야 함을, B라는 것은 파랑으로 칠해야 함을 의미한다. ($1 \leq i \leq 2(N-1)$)

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
4 1 2 1 3 1 4 2 3 2 4 3 4	YES RBBRBR
5 1 5 3 5 3 4 4 5 2 3 2 5 1 2 1 3	YES BRRBBBRR
8 1 2 1 3 2 3 2 4 3 5 4 5 4 6 4 7 4 8 5 6 5 7 5 8 6 7 7 8	NO

노트

세 개의 예제를 그림으로 나타내면 아래와 같다.



문제 J. 문자열 X

시간 제한 3 초
메모리 제한 1024 MB

길이가 1 이상이고 알파벳 소문자로만 이루어진 미지의 문자열 X 가 있다.

이 문자열을 알아내기 위해 N 개의 단서 $S_1, S_2, \dots, S_N$ 을 모았다.

N 개의 단서 중 X 를 부분문자열(substring)로 가지는 것이 정확히 K 개 있음을 알게 되었다.

이때 X 로 가능한 문자열은 몇 개가 있을까?

입력

첫째 줄에 두 정수 N 과 K 가 주어진다. ($1 \leq N \leq 500000, 1 \leq K \leq N$)

이후 N 개 줄에 걸쳐 알파벳 소문자로만 이루어진 문자열 S_i 가 각각 주어진다. ($1 \leq |S_i| \leq 500000$, $|S_1| + |S_2| + \dots + |S_N| \leq 500000$)

출력

첫째 줄에 X 로 가능한 문자열의 개수를 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
5 3 bridgetown georgetown freetown capetown kingstown	2
5 5 bridgetown georgetown freetown capetown kingstown	10
3 2 aaaaa bbbb ccc	0

노트

첫 번째 예제에서 X 로 가능한 문자열은 **g**, **r**가 있다.

두 번째 예제에서 X 로 가능한 문자열은 **t**, **o**, **w**, **n**, **to**, **ow**, **wn**, **tow**, **own**, **town**이 있다.

문자열 A 가 문자열 B 를 부분문자열로 가진다는 것은 A 의 양 끝에서 각각 0개 이상의 문자를 지워 B 를 만들 수 있음을 의미한다.

문제 K. 누텔라 트리 (Hard)

시간 제한 7 초
메모리 제한 1024 MB

민제는 정점이 N 개인 트리를 가지고 있다. 이 트리의 각 정점은 빨간색 또는 검은색으로 칠해져 있다.

민제는 빨간색과 검은색 정점들로 가득한 이 트리를 보고 누텔라(Nutella)를 떠올렸다. 누텔라는 민제가 가장 좋아하는 초콜릿 잼으로, 로고는 다음과 같이 생겼다. 맨 앞 글자는 검은색, 나머지 글자는 빨간색임에 주목하자.



그림 1: 누텔라 로고

민제는 트리에서 누텔라 로고를 몇 개나 찾을 수 있을지 궁금해졌다.

다음 조건들을 만족하는 서로 다른 정점들의 열 $[v_1, v_2, \dots, v_k]$ 를 **누텔라 경로**라 정의하자.

- k 는 2 이상이다.
- 각 $1 \leq i \leq k-1$ 에 대해, v_i 와 v_{i+1} 은 트리에서 간선으로 직접 연결되어 있다.
- v_1 은 검은색이다.
- 각 $2 \leq i \leq k$ 에 대해, v_i 는 빨간색이다.

주어진 트리에서 누텔라 경로가 총 몇 개 있는지 구하시오.

단, 민제는 정점 하나를 골라 색을 바꾸는 작업을 총 Q 회 수행할 예정이다. 해당 정점이 검은색이었다면 빨간색으로, 빨간색이었다면 검은색으로 색이 바뀐다. 이 작업을 수행할 때마다 누텔라 경로의 개수를 새로 구해야 한다.

입력

첫째 줄에 트리의 정점의 개수 N 이 주어진다. ($2 \leq N \leq 100000$)

둘째 줄부터 $(N-1)$ 개 줄에 걸쳐 각 간선이 잇는 두 정점의 번호 u_i, v_i 가 공백을 사이에 두고 주어진다. ($1 \leq u_i \leq N, 1 \leq v_i \leq N, u_i \neq v_i$)

$(N+1)$ 째 줄에 알파벳 **B, R**로만 이루어진 길이 N 의 문자열 C 가 주어진다. C 의 i 번째 문자는 i 번 정점의 색을 나타내며, **B**는 검은색, **R**는 빨간색을 의미한다.

그 다음 줄에 정점의 색을 바꾸는 횟수 Q 가 주어진다. ($1 \leq Q \leq 100000$)

그 다음 줄부터 Q 개 줄에 걸쳐 색을 바꿀 정점의 번호 q_i 가 주어진다. ($1 \leq q_i \leq N$)

출력

총 $(Q+1)$ 개 줄에 걸쳐 정답을 출력한다.

첫째 줄에는 맨 처음 주어진 트리에서의 누텔라 경로의 개수를 출력한다.

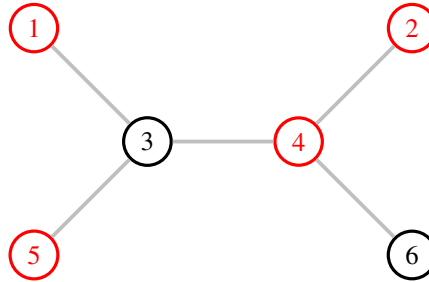
$(i+1)$ 번째 줄에는 색을 바꾸는 작업을 i 회 수행한 이후 누텔라 경로의 개수를 출력한다. ($1 \leq i \leq Q$)

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
6	6
1 3	5
2 4	8
5 3	9
4 6	8
3 4	
RRBRRB	
4	
1	
3	
2	
1	

노트

예제로 주어진 트리의 초기 상태를 그림으로 나타내면 다음과 같다.



1번 정점의 색을 바꾸면 트리는 다음과 같이 변한다.

