UCPC 2022

전국 대학생 프로그래밍 대회 동아리 연합
여름 대회 2022

Preliminaries

Official Problemset

예선 문제

2022년 7월 2일, 14:00 → 17:00

주최

전국 대학생 프로그래밍 대회 동아리 연합

후원

SOLVED.AC
ALGOSPOT
MOLOCO
SAMSUNG SOFTWARE MEMBERSHIP
STARTLINK
NAVER D2
한빛미디어 Hanbit Media, Inc.
FURIOSA
PLANETARIUM
DEVOCEAN
TWIP
programmers
HYUNDAI AutoEver
NEXON
SCVSOFT
정현환 (LIBe)

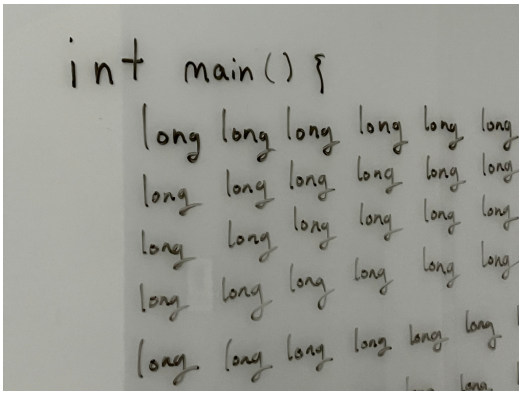
문제 목록

문제지에 있는 문제가 총 10문제가 맞는지 확인하시기 바랍니다.

모든 문제의 메모리 제한은 1GB로 동일합니다.

문제 A. 코딩은 체육과목 입니다

시간 제한 0.5 초 메모리 제한 1024 MB



오늘은 혜아의 면접 날이다. 면접 준비를 열심히 해서 앞선 질문들을 잘 대답한 혜아는 이제 마지막으로 칠판에 직접 코딩하는 문제를 받았다. 혜아가 받은 문제는 두 수를 더하는 문제였다. C++ 책을 열심히 읽었던 혜아는 간단히 두 수를 더하는 코드를 칠판에 적었다. 코드를 본 면접관은 다음 질문을 했다. "만약, 입출력이 N바이트 크기의 정수라면 프로그램을 어떻게 구현해야 할까요?"

혜아는 책에 있는 정수 자료형과 관련된 내용을 기억해 냈다. 책에는 **long int**는 4바이트 정수까지 저장할 수 있는 정수 자료형이고 **long long int**는 8바이트 정수까지 저장할 수 있는 정수 자료형이라고 적혀 있었다. 혜아는 이런 생각이 들었다. "**int** 앞에 **long**을 하나씩 더 붙일 때마다 4바이트씩 저장할 수 있는 공간이 늘어나는 걸까? 분명 **long long long int**는 12바이트, **long long long long int**는 16바이트까지 저장할 수 있는 정수 자료형일 거야!" 그렇게 혜아는 당황하는 면접관의 얼굴을 뒤로한 채 칠판에 정수 자료형을 써 내려가기 시작했다.

혜아가 N바이트 정수까지 저장할 수 있다고 생각해서 칠판에 쓴 정수 자료형의 이름은 무엇일까?

입력

첫 번째 줄에는 문제의 정수 N이 주어진다. ($4 \le N \le 1000$; N은 4의 배수)

출력

혜아가 N바이트 정수까지 저장할 수 있다고 생각하는 정수 자료형의 이름을 출력하여라.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
4	long int
20	long long long long long int

노트

출력에서 **long**과 **long**, **long**과 **int** 사이에는 공백이 하나씩 들어간다.

실제로 C++에서 각 정수 자료형이 저장할 수 있는 수의 크기는 환경에 따라 달라질 수 있다. 덧붙여, 실제로 문제 내용과 같이 **long long long int**와 같은 자료형을 사용한 코드를 GCC의 C++ 컴파일러를 사용해 컴파일하려고 할 경우 `'long long long' is too long for GCC`라는 에러 메시지와 함께 컴파일되지 않는다.

문제 B. N수매화검법

시간 제한 2 초 메모리 제한 1024 MB

화산파의 장로 우경은 새로운 무공 **N수매화검법**을 창안했다. N수매화검법은 이십사수매화검법을 발전시킨 검법으로 총 N개의 베기(검으로 무언가를 베는 동작)로 이루어진다.

N수매화검법은 2차원 평면 상에서 펼치는 검법으로, 베기 i는 점 s_i에서 시작해 e_i까지를 일직선으로 벤다. 이때 검이 지나는 경로를 베기 i의 경로라고 한다.

또한 검법을 펼치는 동안 한 번 벨 때마다 심오한 원리로 내공을 소모하는데, 그 원리란 다음과 같다.

- 베기 i에는 가중치 w_i가 정해져 있으며, 베기 i를 행하면 $(m+1) \times w_i$만큼의 내공을 소모한다. 이때 m은 베기 i를 행한 순간에 아직 행하지 않은 베기 중 베기 i와 경로가 교차하는 베기의 개수다.

N수매화검법을 완성하기 위해서는 검법을 이루는 N개의 베기를 모두 정확히 한 번씩 행해야 하나, 그 순서는 상관이 없다. 장로 우경은 최소한의 내공만을 소모하여 N수매화검법을 완성하고 싶다. 그를 위해 N수매화검법을 완성하는 데 소모해야 하는 내공의 합의 최솟값을 구해주자.

입력

첫 번째 줄에 베기의 개수 N이 주어진다. $(1 \le N \le 2500)$

다음 N개의 줄에는 각 줄마다 베기 i에 대해 s_i, e_i의 좌표 (sx_i, sy_i), (ex_i, ey_i)와 가중치 w_i가 공백으로 구분되어 차례로 주어진다. $(-10^9 \le sx_i, sy_i, ex_i, ey_i \le 10^9;\ 1 \le w_i \le 10^9)$

주어지는 $2N$개 점의 위치는 모두 서로 다르며, 어떤 세 점도 같은 직선 위에 있지 않다.

입력으로 주어지는 모든 수는 정수다.

출력

N수매화검법을 완성하는 데 소모해야 하는 내공의 합의 최솟값을 출력하라.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
3 1 1 3 3 1 1 3 3 1 2 6 7 8 9 1	5

문제 C. 수소철도 충전 시스템

시간 제한 4 초 메모리 제한 1024 MB

탄소중립 시대를 맞아 한국에도 수소철도가 도입되었다. 정부는 UCPC차량사업소를 수소열차를 위한 거점 기지로 지정하여 여기에 수소철도 관련 업무를 모두 맡기려고 한다.

선우는 수소철도 연료 충전 시스템을 담당하여 충전소를 관리하는 임무를 맡았다. 현재 UCPC차량사업소에서 충전소로 사용하는 시설에는 N개의 교차로가 있고 $N-1$개의 레일이 교차로 사이를 트리 형태로 연결하고 있다. 교차로의 번호는 1 이상 N 이하의 서로 다른 정수이다. 모든 레일의 길이는 열차 1량의 길이와 같기 때문에, 충전소에 열차가 주차한다면 특정 두 교차로를 연결하는 단순 경로 위에 열차를 세워 열차의 각 칸(1량)이 레일 하나를 차지하도록 할 수 있다.

모든 레일은 단선으로 설치했기 때문에 한 레일 위에는 최대 한 대의 열차만 주차할 수 있다. 다만, 열차의 칸과 칸 사이를 이어 주는 통로는 고무 재질로 잘 휘기 때문에 교차로에서는 여러 열차가 겹칠 수 있다.

일부 교차로에는 충전기가 설치되어 있다. 열차를 충전하려면 열차의 기관실이 있는 한쪽 끝이 충전기가 있는 교차로에 닿도록 주차해서 충전기와 기관실을 연결해야 한다. 열차마다 제조사나 규격이 모두 다르기 때문에 각 열차는 특정 교차로에 있는 충전기만 사용할 수 있다.

철도 운행 상황에 따라 충전소에 들어오는 열차의 상황이 시시각각 달라지기 때문에 선우는 열차 배치 시스템을 개발하려고 한다. 충전소에 열차가 총 T대가 들어오는데, $j(1 \le j \le T)$번째 열차는 길이가 l_j량이고 반드시 p_j번 교차로에 있는 충전기를 사용해야 한다.

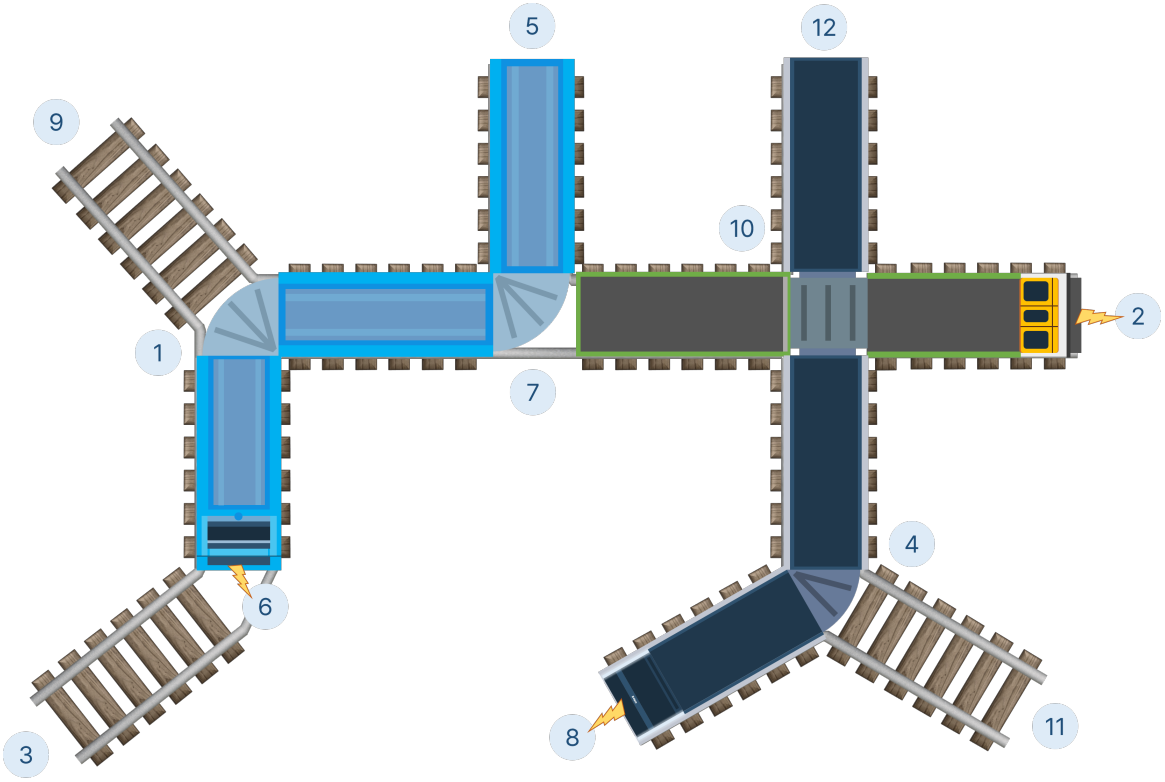


그림 C.1: 첫 번째 예제에서 열차를 배치하는 예시

선우는 여러 열차가 같은 레일을 쓰지 않도록 열차를 배치하는 게 생각보다 어렵다는 것을 파악하고 당신에게 도움을 요청하였다. 충전소와 열차의 정보가 주어지면 적절하게 열차를 배치하는 프로그램을 작성하여라.

입력

첫 번째 줄에는 교차로의 수 N이 주어진다. $(2 \le N \le 500\,000)$

이후 $N-1$줄에 걸쳐 레일들의 정보가 주어진다. 이 중 i번째 줄에는 두 개의 정수 s_i와 e_i가 공백으로 구분되어 주어지며, 이는 i번째 레일이 s_i번 교차로와 e_i번 교차로를 연결함을 의미한다. $(1 \le s_i, e_i \le N)$

주어지는 레일들은 교차로들을 트리 형태로 연결하고 있음이 보장된다.

$N+1$번째 줄에는 충전해야 할 열차의 수 T가 주어진다. $(1 \le T < N)$

이후 T줄에 걸쳐 열차들의 정보가 주어진다. 이 중 j번째 줄에는 두 개의 정수 p_j와 l_j가 공백으로 구분되어 주어지며, 이는 j번째 열차의 충전기가 p_j번 교차로에 있고, 열차의 길이는 l_j량임을 의미한다. $(1 \le p_j \le N;\ 1 \le l_j < N)$

출력

만약 열차 T대를 모두 충전하는 방법이 없다면 첫 번째 줄에 **NO**를 출력한다.

만약 열차 T대를 모두 충전할 수 있다면 첫 번째 줄에 **YES**를 출력하고, 두 번째 줄부터 T개의 줄에 걸쳐 열차의 배치를 출력한다. 이 중 j번째 줄에는 두 개의 정수 p_j와 q_j를 출력해야 하며, 이는 j번째 열차를 p_j번 교차로와 q_j번 교차로를 연결하는 단순 경로 위에 배치함을 의미한다. 이때 이 단순 경로의 길이는 l_j여야 하며, 여기서 출력하는 p_j는 입력으로 주어진 j번째 열차의 충전기의 위치 p_j와 동일해야 한다. $(1 \le p_j, q_j \le N)$

가능한 답이 여러 가지라면 그중 아무거나 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
12 1 6 7 10 4 8 3 6 10 12 5 7 11 4 1 9 2 10 7 1 10 4 3 6 3 8 3 2 2	YES 6 5 8 12 2 7
5 1 2 2 3 2 4 3 5 2 2 2 5 1	NO

문제 D. functionx

시간 제한 1초 메모리 제한 1024 MB

x를 변수로 하는 다항식 $f(x)$가 있다. 처음에 $f(x) = 1$이다. 이때, 다음 쿼리를 수행하는 프로그램을 작성하시오.

- 1 a b: $f(x)$에 $(ax + b)$를 곱한다.
- 2 c: $f(c)$가 음수라면 `-`, 0이라면 `0`, 양수라면 `+`를 출력한다.

입력

첫 번째 줄에 쿼리의 개수 Q가 주어진다. $(1 \le Q \le 200000)$

다음 Q개의 줄에는 쿼리가 문제에서 언급한 형식으로 한 줄에 하나씩 주어진다. 2번 쿼리는 하나 이상 주어진다. $(-10^{18} \le a, b, c \le 10^{18}$; a, b, c는 정수)

출력

2번 쿼리가 주어질 때마다 정답을 한 줄에 하나씩 출력한다.

입출력 예시

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

- 1번째 쿼리 이후에 $f(x) = 2x + 3$이고, $f(-2) = -1 < 0$이다.
- 3번째 쿼리 이후에 $f(x) = (2x + 3) \times (-x - 5) = -2x^2 - 13x - 15$이고, $f(-5) = 0$이다.
- 5번째 쿼리 이후에 $f(x) = (-2x^2 - 13x - 15) \times (7x - 4) = -14x^3 - 83x^2 - 53x + 60$이고, $f(0) = 60 > 0$이다.

문제 E. solved.ac 2022

시간 제한 2 초 메모리 제한 1024 MB

solved.ac는 프로그래밍 문제해결 사이트 백준 온라인 저지에 있는 문제들에 난이도를 붙이는 커뮤니티 프로젝트다. 기존에는 백준 온라인 저지의 문제들에 난이도 표기가 없어서, 다양한 문제를 풀어 보고 싶더라도 난이도를 가늠하기 어려워 무슨 문제를 풀어야 할지 판단하기 곤란했기 때문에 만들어졌다. solved.ac가 생긴 이후 1900명 이상의 기여자 분들께서 소중한 난이도 의견을 공유해 주셨고, 지금은 약 16000문제에 난이도가 붙게 되었다.

어떤 문제의 **난이도**는 그 문제를 푼 사람들이 제출한 난이도 의견을 바탕으로 결정한다. 기존에는 의견이 제출된 시점과 상관 없이 단순 절사평균으로 난이도를 결정했으나, 프로그래밍 문제 해결은 빠르게 변하는 분야이기 때문에 solved.ac는 새로운 의견들을 더 무겁게 반영하고자 난이도 산정 공식을 바꾸기로 했다. 어떤 문제에 N개의 난이도 의견이 제출되었을 때 문제의 난이도는 아래와 같은 방법으로 결정한다.

- 난이도 의견이 하나도 없다면 문제의 난이도는 0으로 한다.
- 난이도 의견이 하나 이상 있는 경우, 의견들을 제출한 시각에 따라 정렬한 뒤 각 의견의 가중치 p_i를 다음 수식으로 결정한다.

$$p_i = \max\left(0.5^{(t_N - t_i)/365\text{일}}, 0.9^{N-i}\right)$$

  - t_i는 i번째 난이도 의견이 제출된 시각을 의미한다. 시간 순으로 정렬했으므로 $t_1 \le t_2 \le \cdots \le t_N$이며, t_N은 가장 최근 제출된 의견의 제출 시각이 된다.
- 문제의 난이도 X는 p_i를 가중치로 갖는 가중평균을 소수점 아래 첫 번째 자리에서 반올림해 정수로 나타낸 것이다.

$$X = \frac{p_1 l_1 + p_2 l_2 + \cdots + p_N l_N}{p_1 + p_2 + \cdots + p_N} = \frac{\sum_{i=1}^{N} p_i l_i}{\sum_{i=1}^{N} p_i}$$

  - l_i는 i번째 의견에 담긴 난이도를 1 이상 30 이하의 정수로 바꾼 값이다. 1은 가장 낮은 난이도인 브론즈 5를, 30은 가장 높은 난이도인 루비 1을 가리킨다.

사용자들이 어떤 문제에 제출한 난이도 의견 목록이 주어질 때, solved.ac가 결정한 문제의 난이도를 계산하는 프로그램을 작성하시오.

입력

첫 번째 줄에 난이도 의견의 개수 N이 주어진다. $(0 \le N \le 1000)$

이후 두 번째 줄부터 다음 N개의 줄에 걸쳐 각 줄마다 난이도 의견이 하나씩 주어진다. 주어지는 N개 줄 중 i번째 줄에는 t_i와 l_i가 공백으로 구분되어 주어진다.

- t_i는 난이도 의견이 남겨진 시각으로, 연도/월/일 시:분:초 형식이다.
  - 연도는 4자리, 월/일/시/분/초는 각각 2자리의 문자열이며 필요한 경우 자릿수를 맞추기 위해 숫자 0이 앞에 붙을 수 있다. 특히 2020년은 윤년이므로, 2020년의 경우 2월의 마지막 날이 29일이다.
  - 시간은 24시간 형식이다.
  - 주어지는 시각은 2019년 6월 6일 00:00:00부터 2022년 7월 1일 23:59:59까지이며, $t_1 \le t_2 \le \cdots \le t_N$이 되도록 정렬되어 있다.
- l_i는 난이도 의견의 값을 의미하는 정수이다. $(1 \le l_i \le 30)$

출력

주어진 난이도 의견들을 바탕으로 계산된 난이도를 출력한다. 출력한 정수가 채점 프로그램이 계산한 반올림하기 전의 가중평균과 $0.5 + 10^{-5} = 0.50001$ 이하의 차이가 발생하는 경우에 정답으로 인정한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
2 2019/07/03 03:43:21 24 2021/10/14 10:12:31 18	21
3 2020/08/01 22:41:52 24 2020/08/02 09:20:19 26 2020/08/15 15:17:53 24	25
0	0

노트

첫 번째 경우에서 첫 번째 의견은 마지막 의견에 비해 약 834일 6시간 ≈ 2.286년 전 의견이므로, 가중평균은

$$X \approx \frac{\max\left(0.5^{2.286}, 0.9^{1}\right) \times 24 + \max\left(0.5^{0}, 0.9^{0}\right) \times 18}{\max\left(0.5^{2.286}, 0.9^{1}\right) + \max\left(0.5^{0}, 0.9^{0}\right)} = \frac{0.9 \times 24 + 1 \times 18}{0.9 + 1} \approx 20.842$$

이다. 따라서 난이도는 이를 반올림한 21로 결정된다.

문제 F. Twitch Plays VIIIbit Explorer

시간 제한 1초 메모리 제한 1024 MB

생방송 플랫폼 Twitch에서 "Twitch Plays Pokémon"이라는 컨텐츠가 큰 유행을 끈 적이 있다. 이 방송은 시청자가 채팅으로 명령을 입력하면 그 명령이 게임에 반영되는 독특한 시스템을 사용해, 수십만 명이 합심해 하나의 게임을 진행한다는 특별한 경험을 제공했다. 막 데뷔한 스트리머 에리는 이 시스템을 빌려 "Twitch Plays VIIIbit Explorer" 컨텐츠를 진행하기로 했다.

VIIIbit Explorer는 캐릭터를 조종해 던전을 탈출하는 간단한 게임이다. 던전은 격자로 나뉘어진 직사각형 모양이며, 몇몇 격자칸에는 아이템이 놓여 있다. 시청자는 다음과 같은 명령들을 입력해 캐릭터를 조종할 수 있다.

- **U**, **D**, **L**, **R**: 캐릭터를 위, 아래, 왼쪽, 오른쪽으로 한 칸 움직인다. 단, 던전 바깥으로 이동하게 되는 경우 던전을 둘러싸고 있는 용암에 빠져 게임 오버된다.
- **P**: 캐릭터가 위치한 칸에 놓여 있는 아이템을 줍는다. 단, 아이템이 없는 칸에서 이 명령을 사용하는 경우 바닥이 무너져 게임 오버된다.

평화롭게 컨텐츠를 진행하던 중, EJN의 Twitch 스트리머 후원 서비스인 Twip으로 미션 하나가 들어왔다.

> 이 미션을 수락하면 캐릭터는 새로운 던전으로 이동하게 됩니다. 던전은 N행 M열 크기이며 캐릭터는 가장 왼쪽 위의 칸에서 출발합니다.
>
> 이 던전의 모든 칸에는 아이템이 하나씩 놓여 있고, 각 아이템에는 알파벳이 하나씩 적혀 있습니다. 던전을 탐험하며 주운 아이템들을 주운 순서대로 나열했을 때, 그 문자열이 정확히 제 아이디와 일치하면 가진 아이템들이 모두 없어지고 캐릭터가 한 단계 강화됩니다.
>
> 던전을 탈출하기 위해서는 숨겨진 포탈을 작동시켜야 합니다. 캐릭터를 던전의 가장 오른쪽 아래 칸에 위치시킨 뒤, "ALL PERFECT"를 외치면 숨겨진 포탈이 작동할 것입니다. 단, 캐릭터가 아이템을 하나라도 가지고 있다면 포탈은 작동하지 않습니다.
>
> 캐릭터를 한 번 강화시킬 때마다 미션 상금으로 1**억 원**씩이 추가됩니다. 행운을 빕니다!

"1억 원?!" 상금에 큰 감명을 받은 에리는 당장 미션을 수락했다. 그리고 방해되는 시청자들을 제거하기 위해 자신만 채팅을 칠 수 있도록 채팅방을 얼려 버렸다.

이제 남은 것은 최대한 많은 상금을 얻으면서 던전을 탈출할 수 있는 이동 방법을 찾고, 한 글자씩 채팅으로 입력하는 것뿐이다. 에리를 도와 주면 상금을 조금 나눠 줄지도 모르니, 그러한 이동 방법을 찾아 주자.

입력

첫 번째 줄에는 던전의 세로 길이 N, 가로 길이 M, 그리고 아이디의 길이 $|S|$가 공백으로 구분되어 주어진다. ($2 \le N, M \le 50$; $1 \le |S| \le 1000$)

두 번째 줄부터 다음 N개의 각 줄에는 영어 소문자로 이루어진 길이 M의 문자열이 주어진다. i번째 문자열의 j번째 글자는 위에서 i번째, 왼쪽에서 j번째 칸에 있는 아이템에 적힌 알파벳을 의미한다.

마지막 $N+2$번째 줄에는 영어 소문자로 이루어진 아이디 S가 주어진다.

출력

첫 번째 줄에 C와 L을 공백으로 구분하여 출력한다. 이는 캐릭터를 총 C번 강화할 수 있고, 탈출을 위해 포탈을 작동시키기 전까지 L번의 행동이 필요함을 의미한다. 이때, C는 가능한 방법 중 가장 큰 값이어야 한다.

두 번째 줄에는 길이 L의 문자열을 출력한다. 이 문자열의 i번째 문자는 i번째로 진행된 행동이어야 하며, 실제로 출력에 따라 행동을 수행했을 때 캐릭터가 C번 강화되었고 "ALL PERFECT"를 외치는 즉시 던전을 탈출할 수 있는 상태여야 한다.

L은 1 이상 1000000 이하여야 하지만, **이동 횟수가 최소일 필요는 없다**.

가능한 모든 입력에 대해서 위 출력 조건을 만족하는 출력이 항상 존재함을 증명할 수 있다.
가능한 답이 여러 가지라면 그중 아무거나 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
3 3 4 icp cuc pci ucpc	1 12 DRPDPLPRRUPD
2 4 3 maxi imax mai	2 18 PRPRRPDLLPRPLLPRRR

문제 G. SCV 체인

시간 제한 1초 메모리 제한 1024 MB

블록체인 컨설팅 회사 SCVSoft에서는 두 대의 건설 로봇(SCV)이 일하고 있다. 두 로봇에게는 "A"와 "B"라는 멋진 이름이 붙어 있다.

어느 날, 심심해진 두 로봇은 블록체인 만들기 놀이를 하기로 했다. 놀이의 규칙은 다음과 같다.

1. $2N$개의 나무 블록에 1번부터 $2N$번까지 번호를 매기고, 각자 N개씩 나눠 갖는다.
2. 로봇 A부터 시작해 번갈아가며 다음의 두 동작 중 하나를 한다.
    - 블록: 가지고 있는 블록 중 하나를 바닥에 놓는다.
    - 체인: 가지고 있는 블록 중 하나를 바로 전에 상대방이 놓은 블록 위에 올려놓는다. 이때, 새로 놓는 블록의 번호는 상대방이 놓은 블록의 번호보다 커야 한다.

    처음 놀이를 시작할 때는 상대방이 놓은 블록이 없으므로, A는 반드시 블록 동작을 한다.
3. 각자 N번의 동작을 해서 가진 블록이 다 떨어지면 놀이가 끝난다.

여기서 블록 동작은 새로운 블록체인의 시작을 본뜬 중요한 동작이기 때문에, 로봇들은 블록 동작을 할 때마다 어떤 블록을 놓았는지 작은 데이터베이스에 기록으로 남긴다. 데이터베이스는 기록들을 요청이 들어온 순서에 따라 안전하게 저장한다. 그러나 로봇과 데이터베이스 사이의 통신은 약간 불안정해서, 기록을 남기다 보면 기록 저장 요청을 보낸 로봇을 혼동하거나 요청을 누락하는 등의 오류가 발생하기도 한다.

로봇들이 놀이를 한 차례 진행한 뒤 데이터베이스에 저장된 기록이 주어진다. 이때 이 기록이 통신에 오류가 없었다고 가정할 때 놀이에서 실제로 나올 수 있는 기록인지 판별하고, 만약 그렇다면 기록에 체인 동작을 적절히 추가해 놀이에서 있었던 모든 동작이 담긴 기록으로 만들어 보자.

입력

첫 번째 줄에 두 개의 정수 N과 M이 공백으로 구분되어 주어진다. M은 데이터베이스에 저장된 블록 동작 기록의 개수이다. ($1 \le N \le 100\,000$; $1 \le M \le 2N$)

이후 M줄에 걸쳐 블록 동작 기록이 한 줄에 하나씩 주어진다. 기록은 `<robot> BLOCK <number>` 꼴으로, `<robot>`에는 동작을 한 로봇의 이름, `<number>`에는 로봇이 놓은 블록의 번호가 들어간다.

주어지는 기록의 순서는 데이터베이스에 저장된 순서를 따른다. 기록의 첫 번째 동작은 반드시 로봇 A의 동작이며, 기록에 등장하는 블록들의 번호는 모두 1 이상 $2N$ 이하의 정수이고 중복되지 않음이 보장된다.

출력

첫 줄에 주어진 블록 동작 기록이 놀이에서 나올 수 있는 기록이라면 `YES`, 그렇지 않다면 `NO`를 출력한다.

`YES`를 출력했을 경우, 이후 $2N$줄에 걸쳐 주어진 기록에 체인 동작을 추가한 전체 동작 기록을 입력과 같은 형식으로 출력한다. 체인 동작은 `<robot> CHAIN <number>` 꼴으로 나타낸다. 주어진 기록으로부터 만들 수 있는 전체 동작 기록이 여러 가지일 경우 그중 아무것이나 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
6 5 A BLOCK 8 A BLOCK 1 B BLOCK 3 B BLOCK 6 A BLOCK 4	YES A BLOCK 8 B CHAIN 9 A BLOCK 1 B CHAIN 2 A CHAIN 10 B BLOCK 3 A CHAIN 5 B BLOCK 6 A CHAIN 7 B CHAIN 11 A BLOCK 4 B CHAIN 12
4 3 A BLOCK 4 B BLOCK 2 A BLOCK 7	NO

문제 H. yo, i herd u liek ternary operators, so..

시간 제한 1초 메모리 제한 1024 MB

진수는 삼항 연산자를 아주 좋아한다. 삼항 연산자는

```
(condition) ? (value_if_true) : (value_if_false)
```

와 같이 생긴 연산자로, `condition`이 참일 경우 `value_if_true`로 평가되고 아닐 경우 `value_if_false`로 평가된다.

진수가 삼항 연산자에서 재미있다고 느끼는 점은 삼항 연산자 안에 또 삼항 연산자를 쓸 수 있다는 것이다. 다시 말해 `condition`과 `value`에도 삼항 연산자가 들어갈 수 있다.

진수는 삼항 연산자를 너무 좋아한 나머지, 연산자가 삼항 연산자밖에 없는 프로그래밍 언어를 개발하기로 했다. 진수가 개발하는 언어의 특징은, 만약 주어진 식이 여러 가지로 해석될 수 있다면 실행할 때마다 다른 해석을 사용해서 다른 결과를 낼 수도 있다는 것이다. 예를 들어 `a?b:c?d:e` 는

```
((a)?(b):(c)) ? (d) : (e)    또는    (a) ? (b) : ((c)?(d):(e))
```

로 해석될 수 있다.

진수는 문득 본인이 작성한 식이 몇 가지 방법으로 해석될 수 있는지 궁금해졌다. 구체적으로 정의하자면,

```
<expr> ::= (<expr>)?(<expr>):(<expr>) | <variable>
<variable> ::= a | b | c | ... | z
```

위의 BNF(Backus-Naur form)에서 `<expr>`로 나타낼 수 있는 문자열 중에서 괄호를 제거하고 비교하였을 때 진수가 작성한 식과 같은 문자열이 되는 경우의 수가 궁금한 것이다.

괄호가 없는 삼항 연산자로만 구성된 식이 주어지면 가능한 해석의 수를 출력하는 프로그램을 작성하자. 해석될 수 있는 방법의 수가 너무 클 수 있으므로, 이 수를 소수인 $1\,000\,000\,007 (= 10^9 + 7)$로 나눈 나머지를 계산한다.

입력

첫 번째 줄에 알파벳 소문자와 `?`와 `:`만으로 이루어진 문자열 S가 주어진다. $(5 \le |S| \le 300\,000)$

주어진 문자열은 적어도 하나 이상의 유효한 식으로 해석될 수 있다.

출력

문제에서 요구하는 수를 $1\,000\,000\,007$로 나눈 나머지를 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
a?b:c?d:e	2
u?c?p:c:h	1

문제 I. 편지 배달

시간 제한 3 초 메모리 제한 1024 MB

UCPC 중학교에서는 다른 반에 있는 친구에게 편지를 보내는 것이 유행이다. UCPC 중학교에는 1반부터 N반까지 총 N개의 반이 있고, 각 반의 교실은 반 번호 순서대로 긴 복도를 따라 나열되어 있다. i반 교실의 위치는 복도의 시작점으로부터의 거리를 나타내는 정수 x_i로 표현된다.

동규는 편지 유행이 좋은 사업 기회라고 생각해서 편지 배달 서비스를 기획했다. 학생들이 앱을 통해 편지 배송 요청을 접수하면, 쉬는 시간에 일괄적으로 편지를 배달해 주는 서비스이다. 각 편지 배송 요청에는 1번부터 M번까지 번호가 붙어 있으며, 출발지와 목적지의 반 번호 쌍 (s_i, e_i)가 적혀 있다.

동규는 원활한 배달을 위해 각 반에서 한 명씩을 뽑아 배달원으로 고용했다. 동규가 배달원들에게 편지 배송 요청을 적절히 배분해 주면, 각 배달원은 그에 따라 편지를 배달하고 동규에게 수당을 받는다. 자세한 규칙은 다음과 같다.

- 각 배달원은 동규가 정해준 순서대로 편지를 배달해야 한다.
- 배달원이 둘 이상의 편지를 지니면 편지가 섞일 수 있으므로 한 번에 하나의 편지만 배달할 수 있다.
- 배달원이 배달을 모두 마치고 나면 수업을 들어야 하기 때문에 다시 자기 반 교실로 돌아와야 한다.
- 각 배달원은 자기 반 교실에서 출발해서 편지를 모두 배달하고 자기 반 교실로 돌아오는 데 필요한 최소 이동 거리만큼 동규에게 수당을 받는다.
- 편지 배송 요청을 배분받지 않은 배달원들은 수당을 받지 않는다.

예를 들어, 4개의 반 교실이 1의 간격을 두고 나란히 있고, 배달해야 하는 편지의 출발지와 목적지 쌍이 순서대로 $(4,2)$, $(1,3)$이라고 하자.

만약 1반의 배달원에게 2번 편지를 배분하고 3반의 배달원에게 1번 편지를 배분한다면, 1반의 배달원이 이동해야 하는 거리는 $2+2=4$이고 3반의 배달원이 이동해야 하는 거리는 $1+2+1=4$이다. 따라서 동규는 두 배달원에게 총 $4+4=8$의 수당을 지급해야 한다. (그림 I.1)

한편, 2, 3, 4반의 배달원들에게는 편지를 배분하지 않고 1반의 배달원에게 2, 1번 편지를 순서대로 배분한다면, 1반의 배달원이 이동해야 하는 거리는 $2+1+2+1=6$이 된다. 따라서 동규는 1반의 배달원에게만 6의 수당을 지급하면 된다. (그림 I.2)

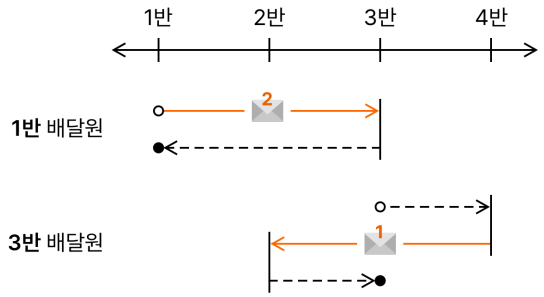


그림 I.1: 동규가 8의 수당을 지급해야 하는 배분 방식

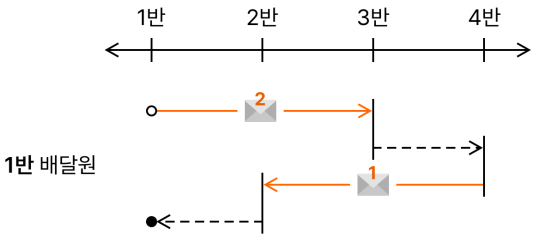


그림 I.2: 동규가 6의 수당을 지급해야 하는 배분 방식

동규는 배달원들에게 지급할 총 수당을 최소화하고 싶다. 위 예시의 경우, 1반의 배달원에게만 2, 1번 편지를 순서대로 배분했을 때 동규가 지급할 총 수당이 최소가 된다. 배달원들에게 편지 배송 요청을 어떻게 배분해야 하는지 구하시오.

입력

첫 번째 줄에 정수 N, M이 공백으로 구분되어 주어진다. $(2 \le N \le 300\,000;\ 1 \le M \le 300\,000)$

두 번째 줄에 N개의 서로 다른 정수 x_i가 공백으로 구분되어 오름차순으로 주어진다. i번째 정수는 i반의 교실 위치를 나타낸다. $(0 \le x_i \le 10^9)$

세 번째 줄부터 다음 M개의 각 줄에 정수 s_i, e_i가 공백으로 구분되어 주어진다. $(1 \le s_i, e_i \le N;\ s_i \ne e_i)$ 이 중 i번째 줄은 i번 편지 배송 요청을 나타내며, s_i는 출발지 교실의 반 번호, e_i는 목적지 교실의 반 번호이다.

출력

첫 번째 줄에 동규가 지급해야 하는 총 수당의 최솟값을 출력한다.

그 다음 N개의 줄을 출력한다. 이 중 i번째 줄에는 i반 배달원이 동규에게 받을 편지 배송 요청의 개수를 출력한 뒤, 동규에게 받을 편지 배송 요청의 번호를 순서대로 출력한다.

가능한 배분 방법이 여러 가지라면 하나만 출력한다.

입출력 예시

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

문제 J. 수 정렬하기, 근데 이제 제곱수를 곁들인

시간 제한 2.5 초 메모리 제한 1024 MB

양의 정수로 이루어진 길이가 N 인 수열 $A_1, A_2, \cdots, A_N$이 존재할 때, 다음 행동을 원하는 만큼 반복할 수 있다.

$1 \le i, j \le N; i \ne j$이면서 $A_i \times A_j$가 제곱수인 i, j를 선택해, A_i와 A_j의 값을 바꾼다.

위 행동을 원하는 만큼 반복하여 수열 $A_1, A_2, \cdots, A_N$을 비내림차순, 즉 $A_1 \le A_2 \le \cdots \le A_N$가 되도록 정렬할 수 있는지 판단하시오.

입력

첫 번째 줄에 N이 주어진다. $(1 \le N \le 500\,000)$

두 번째 줄에 정수 $A_1, A_2, \cdots, A_N$이 공백으로 구분되어 주어진다. $(1 \le A_i \le 10^{18})$

출력

위 행동을 원하는 만큼 반복하여 수열을 비내림차순으로 정렬할 수 있으면 **YES**, 아니면 **NO**를 출력하시오.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
4 1 27 3 12	YES
2 2 1	NO

노트

첫 번째 예시의 경우

- 27과 12의 곱은 324다. $324 = 18^2$이 제곱수이므로 27과 12의 값을 바꿀 수 있고, 수열은 $[1, 12, 3, 27]$이 된다.
- 12와 3의 곱은 36이다. 마찬가지로 $36 = 6^2$이 제곱수이므로 12와 3의 값을 바꿀 수 있고, 수열은 $[1, 3, 12, 27]$이 된다.

$[1, 3, 12, 27]$은 비내림차순으로 정렬된 상태이므로 답은 **YES**가 된다.