



전국 대학생 프로그래밍 대회 동아리 연합
여름 대회 2026

Finals

Official Problemset

본선 문제

2026년 7월 12일, 11:00 - 16:00

주최

전국 대학생 프로그래밍 대회 동아리 연합

후원



ASTEROMORPH



김동현(kdh9949) 김영현(kipa00) 나정휘(jhna917) 정재호(cubic)

문제 목록

문제지에 있는 문제가 총 14문제가 맞는지 확인하시기 바랍니다.

- A** !!New Game Start!!
- B** 11 싫어 수
- C** C4ESAR
- D** 비상 물자 보급
- E** Shift
- F** 배수
- G** Vector Rails
- H** 판다스틱
- I** Yet Another Binary Problem
- J** 무한 문자열
- K** 갑천 자전거길
- L** 아침약
- M** 악령 퇴치
- N** 지그재그

모든 문제의 메모리 제한은 1024 MB로 동일합니다.

문제 A. !!New Game Start!!

시간 제한 4 초 메모리 제한 1024 MB

니나는 최근 발매된 게임, Bitaro's Dungeon을 시작했다. 던전은 -10^{10} 층부터, 10^{10} 층까지 존재하며, 게임을 시작하면 0층에서 시작한다.

게임은 총 N 개의 스테이지로 이루어져 있으며, 스테이지가 바뀔 때마다 층을 이동해 전투하게 된다. $i-1$ 번 스테이지에서 i 번 스테이지로 이동할 때, $A_i = +1$ 이라면 한 층 위로 올라가고, $A_i = -1$ 이라면 한 층 아래로 내려간다.

또한, 게임을 시작하기 전에 M 개의 쿠폰 중 원하는 개수를 구매할 수 있다. Y_i 원을 이용해 i 번 쿠폰을 구매하면, 모든 $L_i \leq t \leq R_i$ 에 대해 다음 효과가 적용된다.

- 스테이지 t 에 진입한 뒤 현재 층수가 X_i 보다 작다면, 현재 층수가 즉시 X_i 층이 된다.

어떤 스테이지 t 에 적용되는 쿠폰이 여러개일 수도 있다. 이 경우 적용되는 쿠폰들 중 X_i 가 가장 큰 쿠폰만이 적용된다.

이 게임에는 **체크포인트** 기능도 존재한다. 현재 x 층에 있을 때, D 원을 내고 체크포인트를 설정하면 그 이후에는 x 층에서 $x-1$ 으로 내려가려고 하면 그대로 x 층에 머무르게 된다. 체크포인트는 여러 번 설정할 수 있다. 게임을 시작한 직후 0층에서 바로 체크포인트를 선언할 수 있음에 유의하라.

니나가 사용할 수 있는 돈은 총 W 원이다. 쿠폰 구매와 체크포인트 설정에 사용한 금액의 합이 W 원을 넘지 않도록 할 때, 모든 스테이지를 진행한 뒤 니나가 있을 수 있는 층수의 최댓값을 구하여라.

입력

첫째 줄에 네 정수 N, M, W, D 가 공백을 사이에 두고 주어진다. 각각 스테이지 수, 쿠폰 개수, 사용할 수 있는 돈, 체크포인트 설정 비용을 의미한다. ($1 \leq N \leq 500000$; $0 \leq M \leq 500000$; $1 \leq W \leq 10^{15}$; $1 \leq D \leq \min(10^9, W)$)

둘째 줄에 N 개의 정수 $A_1, A_2, \dots, A_N$ 이 공백을 사이에 두고 주어진다. ($A_i \in \{-1, 1\}$)

다음 M 개의 줄에 걸쳐 쿠폰의 정보가 주어진다.

i 번째 줄에는 네 정수 L_i, R_i, X_i, Y_i 가 공백을 사이에 두고 주어진다. 이는 각각 쿠폰의 적용 시작 스테이지, 적용 종료 스테이지, 최소 보장 층수, 가격을 의미한다. ($1 \leq L_i \leq R_i \leq N$; $0 \leq X_i \leq N$; $1 \leq Y_i \leq \min(10^9, W)$)

출력

쿠폰 구매와 체크포인트 설정에 사용한 금액의 합이 W 원을 넘지 않도록 할 때, 모든 스테이지를 진행한 뒤 니나가 있을 수 있는 층수의 최댓값을 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
14 3 4 2 1 1 -1 -1 1 1 1 -1 -1 -1 1 1 -1 -1 1 14 1 4 2 10 2 4 8 14 3 4	5
8 3 5 2 -1 -1 1 -1 -1 1 1 -1 1 1 2 3 1 8 1 5 6 8 2 5	3

노트

첫 번째 예제에서는 쿠폰을 구매하지 않고 체크포인트를 2번 설정하는 것이 최적이다. 스테이지 7이 끝난 뒤 3층에 체크포인트를 설정하고, 스테이지 12가 끝난 뒤 5층에 체크포인트를 설정하면 된다.

두 번째 예제에서는 1번 쿠폰을 구매하고 체크포인트를 1번 설정하는 것이 최적이다. 1번 쿠폰의 가격은 3원이고 체크포인트 설정 비용은 2원이므로, 총 5원을 사용한다.

1번 쿠폰을 구매한 뒤, 스테이지 1이 끝난 직후 2층에 체크포인트를 설정하면 된다.

스테이지	층수
1	2
2	2
3	3
4	2
5	2
6	3
7	4
8	3

문제 B. 11 싫어 수

시간 제한 1 초 메모리 제한 1024 MB

어떤 양의 정수 n 에 대해, n 의 숫자 중에서 0개 또는 1개를 0 이상 9 이하의 다른 숫자로 바꾸더라도 n 이 11의 배수가 되지 않는다면, n 을 **11 싫어 수**라고 한다. 이때 맨 앞 자리 숫자를 0으로 바꾸는 것도 허용되며, 0은 11의 배수이다. 양의 정수 A 와 B 가 주어진다. A 이상 B 이하의 11 싫어 수의 개수를 구하시오.

입력

첫째 줄에 정수 A 가, 둘째 줄에 정수 B 가 주어진다. ($1 \leq A \leq B \leq 10^{200000}$)

출력

A 이상 B 이하의 11 싫어 수의 개수를 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
1 314159	2

노트

147은 7을 3으로 바꿔서 11의 배수로 만들 수 있으므로 11 싫어 수가 아니다.

545는 11 싫어 수이다.

문제 C. C4ESAR

시간 제한 1초 메모리 제한 1024 MB

알파벳 대문자와 숫자로만 이루어진 문자열 s 가 주어진다. 당신은 s 에 다음 연산을 원하는 만큼 적용할 수 있다.

- s 에 카이사르 암호를 적용한다.
- s 의 부분 문자열인 $C4$ 를 하나 지운다. 이러한 부분 문자열이 여럿 있을 경우 원하는 것을 지울 수 있다.

카이사르 암호는 문자열의 영어 알파벳 대문자와 숫자를 각각 $A \rightarrow B \rightarrow C \rightarrow \dots \rightarrow Z \rightarrow A$ 와 $0 \rightarrow 1 \rightarrow 2 \rightarrow \dots \rightarrow 9 \rightarrow 0$ 의 규칙에 따라 치환하는 암호이다. 예로 UCPC2026, C4ESAR, ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789에 카이사르 암호를 적용하면 각각 VDQD3137, D5FTBS, BCDEFGHIJKLMNOPQRSTUVWXYZA1234567890가 된다.

부분 문자열은 어떠한 문자열에서 연속한 구간으로 등장하는 문자열이다. 예로 U, PC, UCPC는 UCPC의 부분 문자열이지만, C4ESAR, CU, CC는 UCPC의 부분 문자열이 아니다.

연산을 적절히 수행해서 s 를 빈 문자열로 만들 수 있는지 판별하고, 가능하다면 두 연산을 합해서 최소 몇 번 수행해야 하는지 구해보자.

입력

첫째 줄에 영어 알파벳 대문자와 숫자로만 이루어진 문자열 s 가 주어진다. ($1 \leq |s| \leq 10^6$)

출력

s 를 빈 문자열로 만들기 위해 두 연산을 합해서 최소 몇 번 수행해야 하는지 출력한다. 만약 s 를 빈 문자열로 만들 수 없다면 -1을 대신 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
B3BC43	4
UCPC2026	-1

노트

첫 번째 예제의 경우 다음 순서대로 연산을 진행하면 s 를 빈 문자열로 만들 수 있다.

1. s 의 4번째부터 5번째 글자를 지운다. s 는 B3B3이 된다.
2. s 에 카이사르 암호를 적용한다. s 는 C4C4가 된다.
3. s 의 3번째부터 4번째 글자를 지운다. s 는 C4가 된다.
4. s 의 1번째부터 2번째 글자를 지운다. s 는 빈 문자열이 된다.

두 번째 예제의 경우 연산을 어떻게 진행해도 s 를 빈 문자열로 만들 수 없다.

문제 D. 비상 물자 보급

시간 제한 1 초 메모리 제한 1024 MB

갑작스러운 전쟁으로 UCPC 나라에 위기가 찾아왔다. 국내 항공망이 모두 마비됨에 따라, 정부는 유일한 이동 수단인 육로를 통해 각 도시에 비상 물자를 보급하기로 하였다.

UCPC 나라는 N 개의 도시와 $N-1$ 개의 육로로 이루어져 있다. 각 육로는 두 도시를 양방향으로 연결하며, 육로만을 이용해 임의의 두 도시 사이를 오갈 수 있다.

정부는 현재 총 N 개의 비상 물자를 소유하고 있으며, k 번째 물자는 k 만큼의 가치를 가진다. 보급은 수도인 1번 도시에서 시작되며, 물자를 늦게 수령하는 도시를 배려하기 위해 k 번째로 처음 방문하는 도시에 k 번째 물자를 보급한다. (1번 도시는 첫 번째로 방문한 도시로 간주하여 즉시 1번 물자를 보급받는다.)

구체적인 보급 이동 수칙은 다음과 같다.

- 매 순간, 현재 위치한 도시에서 육로로 연결된 인접한 도시 중 한 곳으로 이동한다.
- 만약 인접한 도시 중 아직 한 번도 방문하지 않은 도시가 존재한다면, 그중 하나를 균등한 확률로 선택하여 이동한다.
- 만약 인접한 모든 도시를 이미 방문한 상태라면, 현재 도시를 처음 방문할 때 거쳐 왔던 직전 도시로 되돌아간다.
- 모든 도시에 물자 보급을 완료하면 즉시 보급 작전을 종료한다.

위와 같은 수칙에 따라 수도를 포함한 모든 도시에 물자를 하나씩 보급하려 할 때, 각 도시가 수령하게 되는 물자 가치의 기댓값을 계산하여라.

입력

첫째 줄에 도시의 개수 N 이 주어진다. ($1 \leq N \leq 200000$)

둘째 줄부터 $N-1$ 개의 줄에 걸쳐 두 정수 a, b 가 공백을 사이에 두고 주어진다. ($1 \leq a, b \leq N; a \neq b$)

이는 a 번 도시와 b 번 도시가 육로로 연결되어 있음을 의미한다.

항상 임의의 두 도시 사이를 이동할 수 있는 경로가 존재함이 보장된다.

출력

N 개의 줄에 걸쳐 정답을 출력한다.

$1 \leq k \leq N$ 에 대해 k 번째 줄에는 k 번 도시가 수령하게 되는 물자 가치의 기댓값을 출력한다.

물자 가치의 기댓값은 항상 유리수임을 보일 수 있다. 기댓값의 기약분수 표현이 $\frac{p}{q}$ 일 때, $q \cdot M \equiv p \pmod{998244353}$ 를 만족하는 정수 M 을 출력해야 한다. ($0 \leq M < 998244353$)

그러한 정수가 유일하게 존재함을 보일 수 있다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
6	1
6 1	4
3 6	499122181
2 5	5
1 5	3
2 4	499122180
8	1
2 6	499122182
8 5	499122182
7 5	499122183
4 5	5
1 6	2
7 6	4
6 3	499122183

노트

첫 번째 예제에서는 두 번째 보급 도시를 선택할 확률이 각각 $\frac{1}{2}$ 이며, 각각의 경우에 대해 도시 간의 보급 순서가 다음과 같이 유일하게 결정된다.

- $1 \rightarrow 6 \rightarrow 3 \rightarrow 5 \rightarrow 2 \rightarrow 4$
- $1 \rightarrow 5 \rightarrow 2 \rightarrow 4 \rightarrow 6 \rightarrow 3$

두 보급 순서에 대해 각각의 도시가 수령하게 되는 물자 가치를 계산해보면 다음과 같다.

도시	순서 1의 물자 가치	순서 2의 물자 가치	물자 가치의 기댓값
1	1	1	$\frac{1}{2} \times 1 + \frac{1}{2} \times 1 = 1$
2	5	3	$\frac{1}{2} \times 5 + \frac{1}{2} \times 3 = 4$
3	3	6	$\frac{1}{2} \times 3 + \frac{1}{2} \times 6 = \frac{9}{2}$
4	6	4	$\frac{1}{2} \times 6 + \frac{1}{2} \times 4 = 5$
5	4	2	$\frac{1}{2} \times 4 + \frac{1}{2} \times 2 = 3$
6	2	5	$\frac{1}{2} \times 2 + \frac{1}{2} \times 5 = \frac{7}{2}$

문제 E. Shift

시간 제한 1 초 메모리 제한 1024 MB

서로 다른 값이 적힌 $N \times N$ 격자판을 생각하자. 두 연산 $\text{RowShift}(r)$ 와 $\text{ColShift}(c)$ 를 다음과 같이 정의한다.

- $\text{RowShift}(r)$: r 행의 각 원소를 오른쪽으로 한 칸씩 밀되, 맨 오른쪽 원소는 맨 왼쪽으로 이동한다.
- $\text{ColShift}(c)$: c 열의 각 원소를 아래쪽으로 한 칸씩 밀되, 맨 아래쪽 원소는 맨 위쪽으로 이동한다.

$(1, 2, \dots, N)$ 의 순열 $(P_1, P_2, \dots, P_N)$ 이 주어질 때, 다음 $2N$ 개의 연산을 순서대로 수행하는 것을 **조작** 한 번이라 한다.

1 번째:	$\text{RowShift}(1)$
2 번째:	$\text{ColShift}(P_1)$
3 번째:	$\text{RowShift}(2)$
4 번째:	$\text{ColShift}(P_2)$
	$\vdots$
$2N - 1$ 번째:	$\text{RowShift}(N)$
$2N$ 번째:	$\text{ColShift}(P_N)$

조작을 충분히 반복하면 격자판이 반드시 처음 상태로 돌아온다. 최소 몇 번 반복해야 하는지 구하여라. 단, 답이 매우 커질 수 있으므로 998 244 353으로 나눈 나머지를 출력하여라.

입력

첫째 줄에 정수 N 이 주어진다. ($1 \leq N \leq 200\,000$)

둘째 줄에 N 개의 정수 $P_1, P_2, \dots, P_N$ 이 공백을 사이에 두고 주어진다. $(P_1, P_2, \dots, P_N)$ 은 $(1, 2, \dots, N)$ 의 순열이다.

출력

조작의 최소 반복 횟수를 998 244 353으로 나눈 나머지를 출력한다.

입출력 예시

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

문제 F. 배수

시간 제한 1 초 메모리 제한 1024 MB

양의 정수 N , M 과 M 개의 쌍 $(P_1, Q_1), (P_2, Q_2), \dots, (P_M, Q_M)$ 이 주어진다. 이때 $P_i < Q_i$ 이다. 양의 정수로 이루어진 수열 $A_1, A_2, \dots, A_N$ 이 다음 두 조건을 모두 만족하면 **좋은 수열**이라고 한다.

- 모든 $1 \leq i \leq N$ 에 대하여, $A_1, A_2, \dots, A_N$ 중 A_i 의 배수인 원소들은 하나의 연속된 구간을 이룬다.
- 모든 $1 \leq i \leq M$ 에 대하여, A_{P_i} 는 A_{Q_i} 의 배수가 아니다.

좋은 수열의 아름다움은 A_i 가 A_j 의 배수인 순서쌍 (i, j) ($1 \leq i, j \leq N$)의 개수로 정의된다.

좋은 수열이 존재하는지 판별하고, 존재한다면 **아름다움**의 최댓값을 구하여라.

입력

첫째 줄에 정수 N , M 이 공백을 사이에 두고 주어진다. ($2 \leq N \leq 100$; $1 \leq M \leq 100$)

다음 M 개의 줄에 걸쳐 정수 P_i, Q_i 가 공백으로 구분되어 주어진다. ($1 \leq P_i < Q_i \leq N$)

출력

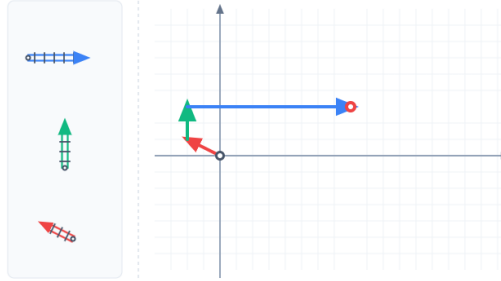
좋은 수열이 존재하지 않으면 -1을 출력한다. 존재한다면 **아름다움**의 최댓값을 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
2 1 1 2	3
표준 입력(stdin)	표준 출력(stdout)
4 1 2 3	13
표준 입력(stdin)	표준 출력(stdout)
4 2 1 3 2 4	12

문제 G. Vector Rails

시간 제한 1 초 메모리 제한 1024 MB



최근 민규는 ‘Vector Rails’라는 철도 건설 게임에 빠져 있다. 이 게임에서는 기점 $(0,0)$ 에서 출발하여, 평면 곳곳에 놓인 여러 목표 도시 각각까지 최소 비용으로 철도를 건설해야 한다.

철도 건설에는 N 개의 건설 모드를 사용할 수 있으며, i 번 건설 모드를 선택해 단위 비용을 투입하면 2차원 평면 위에서 벡터 $\vec{v}_i = (a_i, b_i)$ 의 방향과 길이만큼 선로가 연장된다. 이 벡터는 해당 건설 모드의 ‘진행 방향’과 ‘비용 대비 이동 거리’를 동시에 나타낸다.

철도 건설은 다음 규칙을 따른다.

- 선로 건설은 항상 기점인 $(0,0)$ 에서 시작한다.
- 임의의 건설 모드 i 를 선택하여 원하는 비용 t 만큼 투입해 선로를 연장할 수 있다. 이때 투입하는 비용 t 는 0 이상의 임의의 실수이다.
- 선로를 연장하는 도중 언제든지 다른 건설 모드로 전환하여 이어서 건설할 수 있으며, 건설 모드의 변경 횟수나 순서에는 제한이 없다.

Q 개의 목표 도시 좌표가 주어질 때, 각 목표 도시에 대해 독립적으로 기점에서 그 도시까지 철도를 연결하기 위한 최소 건설 비용을 구하는 프로그램을 작성하시오.

입력

첫째 줄에 건설 모드의 개수 N 과 목표 도시의 개수 Q 가 공백을 사이에 두고 주어진다. ($1 \leq N, Q \leq 200\,000$)

둘째 줄부터 N 개의 줄에 걸쳐, i 번 건설 모드의 방향과 벡터 $\vec{v}_i = (a_i, b_i)$ 를 이루는 두 정수 a_i 와 b_i 가 공백을 사이에 두고 주어진다. ($-10^9 \leq a_i, b_i \leq 10^9$; $(a_i, b_i) \neq (0,0)$)

이어서 Q 개의 줄에 걸쳐, 목표 도시의 좌표 p_x 와 p_y 가 공백을 사이에 두고 주어진다. ($-10^9 \leq p_x, p_y \leq 10^9$)

주어지는 건설 모드의 벡터는 서로 같을 수 있으며, 목표 도시가 출발 기점인 $(0,0)$ 인 경우도 주어질 수 있다.

출력

Q 개의 줄에 걸쳐, 각 목표 도시에 선로를 도달시키기 위한 최소 건설 비용을 출력한다.

어떤 건설 모드를 조합해도 해당 도시에 도달할 수 없는 경우 -1을 대신 출력한다.

도달 가능한 경우 정답을 실수 형태로 출력하며, 정답과의 절대 오차 또는 상대 오차가 10^{-9} 이하이면 정답으로 인정한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
3 4	1.5
4 0	2
0 3	-1
-2 1	0
2 3	
-4 2	
2 -1	
0 0	
4 3	5
1 1	3
-1 1	3
-1 -1	
1 -1	
0 5	
-3 -1	
2 -3	

노트

예제 1의 세 건설 모드의 벡터는 $\vec{v}_1 = (4, 0)$, $\vec{v}_2 = (0, 3)$, $\vec{v}_3 = (-2, 1)$ 이다.

- (2,3): 1번 건설 모드에 비용 0.5를 투입하여 (2,0)에 도달한 뒤, 2번 건설 모드로 전환하여 비용 1을 투입하면 (2,3)에 도달한다. 총 건설 비용은 1.5이다.
- (-4,2): 3번 건설 모드에 비용 2를 투입하면 바로 (-4,2)에 도달한다. 다른 건설 모드로 전환하지 않고 완료할 수 있으며, 총 건설 비용은 2이다.
- (2,-1): 모든 건설 모드의 y 성분이 0 이상이므로, 남쪽 방향으로선로를 연장할 방법이 없다. 따라서 도달할 수 없다.
- (0,0): 기점 자체이므로 총 건설 비용은 0이다.

예제 2의 네 건설 모드의 벡터는 동북, 서북, 서남, 동남 사방을 향해 있어 모든 방향으로 건설할 수 있다.

- (0,5): 1번 건설 모드에 비용 2.5를 투입해 (2.5,2.5)에 도달하고, 2번 건설 모드에 비용 2.5를 투입하면 (0,5)에 도달한다. 총 건설 비용은 5이다.
- (-3,-1): 2번 건설 모드에 비용 1을 투입해 (-1,1)에 도달하고, 3번 건설 모드에 비용 2를 투입하면 (-3,-1)에 도달한다. 총 건설 비용은 3이다.
- (2,-3): 4번 건설 모드에 비용 2.5를 투입해 (2.5,-2.5)에 도달하고, 3번 건설 모드에 비용 0.5를 투입하면 (2,-3)에 도달한다. 총 건설 비용은 3이다.

문제 H. 판다스틱

시간 제한 1 초 메모리 제한 1024 MB



UCPC 동물원에는 N 마리의 판다가 살고 있다. 이곳의 판다들은 립스틱, 치즈스틱, 조이스틱의 세 가지 스틱 중 하나 이상의 스틱을 좋아한다. 동물 애호 단체 UCPC animals는 판다들을 위해 립스틱 A 개, 치즈스틱 B 개, 조이스틱 C 개로 이루어진 총 N 개의 스틱이 담긴 꾸러미를 가져왔다. 이들은 다음 규칙에 따라 스틱을 나누어 주기로 하였다.

1. 판다를 원하는 순서대로 일렬로 줄 세운다.
2. 꾸러미에는 스틱이 차곡차곡 쌓여 있으며, 맨 위에 존재하는 스틱부터 순서대로 하나씩 꺼내 맨 앞에 있는 판다에게 전달한다.
3. 판다는 받은 스틱이 자신이 좋아하는 스틱이라면 그대로 들고 줄을 이탈한다. 그렇지 않다면 뒤에 있는 판다에게 스틱을 전달한다. 이 과정은 어떤 판다가 줄을 이탈하거나 줄의 마지막에 있는 판다가 스틱을 전달받을 때까지 반복된다.
4. 줄의 마지막에 있는 판다가 자신이 좋아하지 않는 스틱을 받을 경우, 스틱 분배 행사는 즉시 실패로 종료된다.

UCPC animals는 꾸러미 안에 스틱들이 어떤 순서로 쌓여 있는지는 알지 못한다. 어떤 순서대로 스틱을 꺼내더라도 무사히 모든 판다에게 스틱을 나누어 줄 수 있는 줄 세우기 방법이 있는지 판별하고, 만약 존재한다면 그런 방법 중 아무거나 하나를 알아내어 UCPC animals가 행사를 성공적으로 마무리하도록 도와주자!

입력

첫째 줄에 테스트 케이스의 개수 T 가 주어진다. ($1 \leq T \leq 10000$)

이후 각 테스트 케이스마다 다음과 같은 형식으로 입력이 주어진다.

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

둘째 줄에 세 스틱의 개수 A, B, C 가 공백을 사이에 두고 주어진다. ($0 \leq A, B, C \leq N$; $A + B + C = N$)

셋째 줄부터 N 개의 줄에 걸쳐 판다들의 정보가 주어진다. 그중 i 번째 줄에는 i 번째 판다가 좋아하는 스틱 종류의 개수 K_i 와 K_i 개의 문자가 공백을 사이에 두고 주어진다. ($1 \leq K_i \leq 3$)

각 문자는 a, b, c 중 하나이며, 각각 립스틱, 치즈스틱, 조이스틱을 의미한다. 한 판다의 정보에서 같은 문자가 두 번 이상 주어지지 않는다.

모든 테스트 케이스에 대한 N 의 합은 500000을 넘지 않는다.

출력

각 테스트 케이스마다, 어떤 순서대로 스틱을 꺼내더라도 모든 판다에게 스틱을 나누어 줄 수 있는 줄 세우기 방법이 존재한다면 한 줄에 Yes를, 그렇지 않다면 No를 출력한다.

그러한 방법이 존재한다면, 다음 줄에 판다를 맨 앞부터 줄 세우는 순서대로 1 이상 N 이하의 서로 다른 N 개의 정수를 공백으로 구분하여 출력한다. 가능한 방법이 여러 가지라면, 아무거나 하나를 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
2	Yes
3	3 1 2
1 1 1	No
2 a b	
3 c a b	
1 b	
3	
3 0 0	
2 a b	
3 c a b	
1 b	

노트

첫 번째 테스트 케이스에서는 판다를 3, 1, 2번 순서대로 세우면 어떤 순서대로 스틱을 꺼내더라도 모든 판다에게 스틱을 나누어 줄 수 있다.

두 번째 테스트 케이스의 꾸러미에는 치즈스틱이 들어있지 않으므로, 세 번째 판다가 원하는 스틱을 받을 수 있는 방법은 존재하지 않는다.

문제 I. Yet Another Binary Problem

시간 제한 5 초 메모리 제한 1024 MB

0과 1로만 이뤄진 $N \times M$ 크기의 격자 A 가 주어진다. 당신은 아래 두 종류의 시행을 원하는 만큼 할 수 있다.

- 같은 행에 있는 두 수 0, 1을 하나씩 없앤다.
- 같은 열에 있는 두 수 0, 1을 하나씩 없앤다.

수가 없어진 칸은 빈 칸이 된다. 시행횟수를 최대로 하는 방법을 하나 찾아보자.

입력

첫째 줄에 두 정수 N, M 이 공백을 사이에 두고 주어진다. ($1 \leq N, M \leq 5000$)

다음 N 줄에 걸쳐 $i+1$ 번째 줄에 격자 A 를 나타내는 M 개의 수 $A_{i1}, A_{i2}, \dots, A_{iM}$ 가 공백없이 주어진다. A_{ij} 는 격자의 i 행 j 열에 적힌 수를 나타낸다. ($0 \leq A_{ij} \leq 1$)

출력

N 줄에 걸쳐 각 수가 어떤 시행에 의해 없어졌는지 나타내는 M 개의 문자 $B_{i1}, B_{i2}, \dots, B_{iM}$ 를 공백없이 출력한다. 이때 i 행 j 열의 수가 1번 시행에 의해 없어진 경우 $B_{ij} = 1$, 2번 시행에 의해 없어진 경우 $B_{ij} = 2$, 없어지지 않은 경우 $B_{ij} = x$ 로 한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
3 4	1212
0111	1111
1001	1212
1000	
2 2	2x
00	2x
10	

문제 J. 무한 문자열

시간 제한 6초 메모리 제한 1024MB

문자열 S 와 Q 개의 구간이 주어진다. 구간은 입력으로 주어진 순서대로 1번부터 Q 번까지 번호가 매겨져 있으며, i 번 구간은 두 정수 l_i, r_i 로 나타낸다.

문자열 S 의 문자는 왼쪽부터 차례대로 1번부터 $|S|$ 번까지 번호가 매겨져 있다. $1 \leq l \leq r \leq |S|$ 에 대해, $S[l, r]$ 은 S 의 l 번째 문자부터 r 번째 문자까지를 순서대로 이어 붙인 부분문자열을 의미한다. 각 i 에 대해 $X_i = S[l_i, r_i]$ 라고 하자.

문자열 X 의 무한문자열 X^∞ 은 X 를 무한히 이어 붙여서 만든 무한 길이의 문자열이다. 즉, $X^\infty = X \cdot X \cdot X \cdots$ 이다.

$X_1^\infty, X_2^\infty, \dots, X_Q^\infty$ 를 사전순으로 오름차순 정렬한 결과를 출력하여라.

단, 두 무한문자열이 완전히 같은 경우에는 더 번호가 작은 구간이 앞에 오도록 한다. 즉, $X_i^\infty = X_j^\infty$ 이고 $i < j$ 라면 i 가 j 보다 앞에 와야 한다.

입력

첫째 줄에 알파벳 소문자로만 이루어진 문자열 S 가 주어진다. ($2 \leq |S| \leq 150\,000$)

둘째 줄에 구간의 개수 Q 가 주어진다. ($1 \leq Q \leq 1\,000\,000$)

이후 Q 개의 줄에 걸쳐 각 구간을 나타내는 두 정수 l_i, r_i 가 공백을 사이에 두고 주어진다. ($1 \leq l_i \leq r_i \leq |S|$)

출력

정렬된 순서대로 구간의 번호를 공백으로 구분하여 출력한다.

구간의 번호는 입력으로 주어진 순서대로 $1, 2, \dots, Q$ 이다.

입출력 예시

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

표준 입력(stdin)

표준 출력(stdout)

ababababca	5 10 1 2 9 8 3 6 7 4
10	
1 4	
1 6	
2 5	
9 10	
1 1	
2 2	
8 9	
7 10	
3 9	
10 10	

노트

첫 번째 예제에서 주어진 부분문자열은 $X_1 = ab$, $X_2 = aba$, $X_3 = ab$, $X_4 = bac$ 이다.

각 부분문자열의 무한문자열을 앞 6자리까지 나타내면 다음과 같다.

- $X_1^\infty = ababab\dots$
- $X_2^\infty = abaaba\dots$
- $X_3^\infty = ababab\dots$
- $X_4^\infty = bacbac\dots$

이를 사전순으로 비교하면 $X_2^\infty < X_1^\infty = X_3^\infty < X_4^\infty$ 이다.

따라서 정렬 결과는 2,1,3,4 이다.

X_1^∞ 과 X_3^∞ 은 완전히 같으므로, 더 번호가 작은 1번 구간이 3번 구간보다 앞에 온다.

문제 K. 갑천 자전거길

시간 제한 1.5 초 메모리 제한 1024 MB

재원이는 기분 전환을 위해 갑천에 산책을 갔다. 갑천에는 강변을 따라 자전거길이 일직선으로 나 있다. 자전거길에는 N 대의 자전거가 각각 등속도로 이동하고 있다.

재원이가 산책을 시작한 시각을 0, 산책을 끝낸 시각을 T 라고 하자. i 번째 자전거의 좌표는 재원이가 산책을 시작했을 때 s_i 이고, 재원이가 산책을 끝냈을 때 e_i 이다. 모든 자전거에 대해 s_i 는 서로 다르고, e_i 도 서로 다르다.

재원이가 산책하는 동안 각각의 자전거는 K 대 이하의 자전거만을 만난다. 두 자전거의 좌표가 같아질 때 두 자전거는 서로를 만난다고 간주한다.

재원이는 임의의 시각 t 에 임의의 좌표 x 로부터 가장 가까운 자전거까지의 거리가 얼마인지 계산하고 싶어졌다. 재원이를 도와 Q 개의 쿼리를 해결해 보자.

입력

첫째 줄에 정수 N, T, K 가 공백을 사이에 두고 주어진다. N 은 자전거의 수, T 는 재원이가 산책을 끝낸 시각, K 는 각 자전거가 다른 자전거를 만난 횟수의 상한이다. ($1 \leq N \leq 1000000$; $1 \leq T \leq 10^9$; $1 \leq K \leq 1000$)

다음 N 개의 줄에 자전거의 처음 좌표 s_i 와 마지막 좌표 e_i 가 공백을 사이에 두고 주어진다. ($0 \leq s_i, e_i \leq 10^9$) 모든 i 에 대해 s_i 는 서로 다르고, e_i 도 서로 다르다.

다음 줄에 쿼리의 개수를 나타내는 정수 Q 가 주어진다. ($1 \leq Q \leq 10000$)

다음 Q 개의 줄에 정수 t, x 가 공백을 사이에 두고 주어진다. 시각 t 에 좌표 x 로부터 가장 가까운 자전거까지의 거리를 묻는 쿼리를 나타낸다. ($0 \leq t \leq T$; $0 \leq x \leq 10^9$)

주어진 입력에서 각각의 자전거는 K 대 이하의 자전거를 만남이 보장된다.

출력

각 쿼리에 대한 답을 기약 분수 p/q 형식으로 한 줄에 하나씩 출력한다. p 는 음이 아닌 정수, q 는 양의 정수이다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
6 12 2	7/1
40 100	0/1
0 50	10/1
110 180	1/2
10 0	73/3
130 160	31/6
70 20	
6	
0 33	
6 45	
12 170	
3 57	
8 181	
5 44	

문제 L. 아침약

시간 제한 1 초 메모리 제한 1024 MB



수빈이는 병원에서 처방을 받고 약국에서 N 일치 약을 구매했다. 하루치 약은 아침약, 점심약, 저녁약 각 한 봉지로 구성되어 있으며, 총 $3N$ 개의 약봉지가 다음 순서대로 일렬로 연결되어 있다.

아침약, 점심약, 저녁약, 아침약, 점심약, 저녁약, ...

일반적인 사람이라면 매일 세 끼를 모두 챙겨 먹으면서 약봉지를 앞에서부터 차례대로 잘라 먹으면 된다. 하지만 수빈이의 생활 패턴은 일반적이지 않다. 어떤 날은 저녁만 먹기도 하고, 어떤 날은 아침을 건너뛰기도 한다. 그래서 수빈이는 약을 앞에서부터 순서대로 먹지 않고, 자신의 식사 순서에 맞추어 먹으려고 한다. 물론 약은 하나도 남기지 않고 모두 먹을 생각이다.

그런데 수빈이는 연결된 약봉지가 두 덩어리로 나뉘는 것이 너무 보기 싫다. 따라서 수빈이는 약을 먹을 때마다 현재 남아 있는 약봉지의 맨 앞 또는 맨 뒤에서만 약봉지를 하나 잘라 먹는다. 중간에 있는 약봉지를 잘라서 남아 있는 약봉지가 두 덩어리로 나뉘게 하는 것은 허용되지 않는다.

수빈이가 약을 먹으려는 순서가 주어질 때, 위 조건을 지키면서 모든 약을 그 순서대로 먹을 수 있는지 판별하라.

입력

첫째 줄에 구매한 약이 며칠치인지 나타내는 정수 N 이 주어진다. ($1 \leq N \leq 1000000$)

둘째 줄에 수빈이가 약을 먹으려는 순서를 나타내는 길이 $3N$ 의 문자열 S 가 주어진다. 문자열 S 는 B, L, D로만 이루어져 있으며, 세 문자는 각각 정확히 N 번 등장한다.

문자 B는 아침약, L은 점심약, D는 저녁약을 의미한다.

출력

조건을 지키면서 모든 약을 S 의 순서대로 먹을 수 있다면 Yes를, 그렇지 않다면 No를 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
1 LBD	No
2 BLDDBL	Yes

노트

첫 번째 예제에서 처음 남아 있는 약봉지는 앞에서부터 아침약, 점심약, 저녁약 순서이다. 처음 먹어야 하는 약은 점심약이지만, 점심약은 맨 앞도 맨 뒤도 아니므로 조건을 지키면서 먹을 수 없다.

두 번째 예제에서는 맨 앞, 맨 앞, 맨 앞, 맨 뒤, 맨 앞, 맨 앞의 순서로 약봉지를 잘라 먹으면 주어진 순서대로 모든 약을 먹을 수 있다.

문제 M. 악령 퇴치

시간 제한 2 초 메모리 제한 1024 MB

신주쿠의 어느 거리에 악령이 스며들었다. 이 거리는 왼쪽 끝에 있는 입구와 오른쪽 끝에 있는 출구를 잇는 선분 모양이며, 두 끝점 사이 거리는 $10^9 + 1$ 이다.

N 명의 주술사가 거리 위에 있다. 주술사들은 입구와 가까운 순서대로 1 이상 N 이하의 정수 번호를 갖고 있다. i 번 주술사는 입구로부터 X_i (X_i 는 양의 정수)만큼 오른쪽에 있으며, 모든 주술사는 서로 다른 위치에 있다. 또한 각 주술사는 빨강 또는 파랑 중 하나의 색을 가진다.

각 저주는 두 정수 a, b 로 주어진다. 악령은 a 번과 b 번 주술사에게 저주를 걸며, 항상 $1 \leq a < b \leq N$ 을 만족한다. 이때 두 주술사를 잇는 선분 위에 저주 영역이 생성되며, 저주 영역은 구간 $[X_a, X_b]$ 이다.

저주 영역이 생성되면, 주술사들은 저주 영역 위의 한 점을 골라 **핵심 위치**로 정한다. 핵심 위치의 좌표는 실수일 수 있다. 핵심 위치를 c 라고 하자. 핵심 위치는 주술사가 있는 위치와 같아도 된다.

핵심 위치가 정해지면 저주 영역은 다음 규칙에 따라 빨간 퇴마 영역과 파란 퇴마 영역으로 나뉜다.

각 $a \leq i < b$ 에 대해, 인접한 두 주술사 사이의 구간 $[X_i, X_{i+1}]$ 을 생각하자.

- $X_{i+1} \leq c$ 이면, 구간 $[X_i, X_{i+1}]$ 전체는 i 번 주술사의 색을 가진 퇴마 영역이 된다.
- $c \leq X_i$ 이면, 구간 $[X_i, X_{i+1}]$ 전체는 $i+1$ 번 주술사의 색을 가진 퇴마 영역이 된다.
- $X_i < c < X_{i+1}$ 이면, 구간 $[X_i, c]$ 는 i 번 주술사의 색을 가진 퇴마 영역이 되고, 구간 $[c, X_{i+1}]$ 는 $i+1$ 번 주술사의 색을 가진 퇴마 영역이 된다.

즉, 핵심 위치 c 를 기준으로 왼쪽에 있는 부분은 왼쪽 끝점에 있는 주술사의 색을 따르고, 오른쪽에 있는 부분은 오른쪽 끝점에 있는 주술사의 색을 따른다.

위 규칙에 의해 저주 영역의 모든 지점은 퇴마 영역으로 덮이며, 서로 다른 퇴마 영역은 끝점에서만 만날 수 있다.

핵심 위치가 c 일 때, 빨간 퇴마 영역의 총 길이를 $R(c)$, 파란 퇴마 영역의 총 길이를 $B(c)$ 라 하자. 이 퇴마 의식의 **부조화도**는 두 값의 차이의 절댓값, 즉 $|R(c) - B(c)|$ 이다.

퇴마 의식의 성공률을 높이기 위해 주술사들은 핵심 위치 c 를 적절히 골라 부조화도를 최소화하려고 한다.

주술사의 정보와 악령이 저주를 거는 정보가 주어지면, 각 저주에 대한 퇴마 의식의 최소 부조화도를 구하는 프로그램을 작성하여라.

입력

첫째 줄에는 주술사의 수 N 과 저주의 수 Q 가 공백을 사이에 두고 주어진다. ($2 \leq N \leq 200\,000$; $1 \leq Q \leq 200\,000$)

둘째 줄부터 N 개의 줄에는 주술사의 좌표 X_i 와 색 C_i 가 공백을 사이에 두고 주어진다. ($1 \leq X_1 < X_2 < \dots < X_N \leq 10^9$, $C_i = R \text{ or } B$)

그 다음 줄부터 Q 개의 줄에는 저주의 내용 a, b 가 공백을 사이에 두고 주어진다. a 와 b 는 악령이 저주를 거는 두 주술사의 번호이다. ($1 \leq a < b \leq N$)

출력

각 저주마다 한 줄에 하나씩, 해당 저주에 대한 최소 부조화도를 출력한다. 최소 부조화도는 항상 정수임을 증명할 수 있다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
4 2	0
1 R	2
6 B	
8 B	
12 R	
1 4	
2 3	

노트

저주 1: 핵심 위치를 $c = 2.5$ 로 정하면, 구간 $[1, 2.5]$ 와 $[8, 12]$ 는 빨간 퇴마 영역이 되고, 구간 $[2.5, 6]$ 와 $[6, 8]$ 은 파란 퇴마 영역이 된다. 빨간 퇴마 영역의 총 길이와 파란 퇴마 영역의 총 길이가 모두 5.5이므로 부조화도는 0이다. 핵심 위치를 $c = 11.5$ 로 정해도 부조화도가 0이 된다.

저주 2: 핵심 위치를 $c = 8$ 로 정하면, 구간 $[6, 8]$ 전체가 파란 퇴마 영역이 된다. 저주 영역에 빨강 주술사가 없으므로 부조화도를 2보다 작게 할 수 없다.

문제 N. 지그재그

시간 제한 1 초 메모리 제한 1024 MB

지그재그 문자열이란 0과 1이 번갈아 나오는 이진 문자열을 뜻한다. 예를 들어 0, 101, 010101은 지그재그 문자열이지만, 00, 0110은 지그재그 문자열이 아니다.

어떤 이진 문자열 A 에 대해, $f(A)$ 를 A 를 지그재그 문자열인 부분 수열들로 분할하기 위해 필요한 지그재그 문자열의 최소 개수라고 정의한다. 예를 들어, 11000101은 다음과 같이 세 개의 지그재그 문자열인 부분 수열 10, 101, 010으로 분할할 수 있다. 첫 행은 원래의 문자열을 나타내고, 아래의 세 행은 세 부분 수열을 나타낸다.

A	1	1	0	0	0	1	0	1
지그재그 문자열 1		1		0				
지그재그 문자열 2	1				0			1
지그재그 문자열 3			0			1	0	

또한, 11000101을 두 개 이하의 지그재그 문자열인 부분 수열로 분할하는 방법이 없음을 보일 수 있다. 따라서 $f(11000101) = 3$ 이다.

길이 N 의 이진 문자열 A 가 주어질 때, A 의 모든 부분 문자열 s 에 대해 $f(s)$ 의 합을 구하시오.

엄밀히는, $1 \leq l \leq r \leq N$ 인 모든 (l, r) 에 대해 A 의 l 번째 문자부터 r 번째 문자까지로 이루어진 문자열 $A_l A_{l+1} \dots A_r$ 을 $A[l, r]$ 로 표기하자. 이때 $\sum_{l=1}^N \sum_{r=l}^N f(A[l, r])$ 의 값을 구하시오.

입력

첫째 줄에는 A 의 길이를 나타내는 정수 N 이 주어진다. ($2 \leq N \leq 300000$)

둘째 줄에는 길이 N 의 이진 문자열 A 가 주어진다.

출력

문제의 답을 정수로 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
4 1101	13
8 10000101	79