

문제 1. 샐러드바

입력 파일: standard input
출력 파일: standard output
시간 제한: 3 seconds
메모리 제한: 128 megabytes

범수는 샐러드바에 갔다. n 개의 과일들이 일렬로 샐러드바에 있었다. 과일들은 사과나 오렌지이다. 범수는 샐러드를 만들 때 연속된 아무 구간이나 샐러드로 만들 수 있다.

그가 샐러드를 만들 때 고르는 과일은 왼쪽에서부터 오른쪽 혹은 오른쪽에서 왼쪽으로 샐러드에 추가된다. 범수는 오렌지를 좋아하기 때문에, 그가 만드는 샐러드에는 특별한 조건이 있다. 샐러드를 만드는 도중에, 왼쪽에서 오른쪽으로 만들든 오른쪽에서 왼쪽으로 만들든 오렌지의 수가 사과의 수 보다 적어서는 안된다. 범수가 조건을 만족하면서 샐러드로 만들 수 있는 가장 긴 구간을 구해주자.

입력 형식

첫째 줄에는 과일의 수를 의미하는 정수 n 이 주어진다. ($1 \leq n \leq 1,000,000$) 다음 줄에는 n 개의 문자로 이루어진 문자열 a_1, a_2, \dots, a_n 이 주어진다. ($a_j \in \{j, p\}$) $a_i = j$ 이면 i 번째 과일이 사과이고, $a_i = p$ 이면 i 번째 과일은 오렌지이다.

출력 형식

첫째 줄에 범수가 조건을 만족하면서 샐러드로 만들 수 있는 가장 긴 구간의 길이를 출력한다. 답이 0일 수 있음에 유의하여라.

서브태스크 1 (20 점)

- $n \leq 1,000$

서브태스크 2 (30 점)

- $n \leq 10,000$

서브태스크 3 (50 점)

추가 제한조건이 없다.

예제

standard input	standard output
6 jpjppj	4

참고 사항

가장 왼쪽과 오른쪽에 있는 사과를 제외하고 나머지 모든 과일들을 샐러드로 만들면 된다.

문제 2. 호텔

입력 파일: standard input
출력 파일: standard output
시간 제한: 3 seconds
메모리 제한: 64 megabytes

지구이웨에는 n 개의 마을이 있고, $n-1$ 개의 도로로 연결되어 있다. 도로 하나는 두 마을을 직접 잇는다. 모든 도로는 같은 길이를 가지고 양방향 통행이 가능하다. 모든 마을에서 다른 마을까지 도로를 통해 이용하는 것이 가능하다. 즉 도로망은 트리 모양이다.

지구이웨의 왕인 범수는 세계의 여행자들을 끌어들이기 위한 세계의 럭셔리 호텔을 지으려고 한다. 범수는 호텔이 다른 마을에 있고, 서로가 서로로부터 같은 거리만큼 떨어져 있기를 원한다. 범수가 세 호텔을 세울 수 있는 경우의 수를 구하여라.

입력 형식

첫째 줄에는 마을의 수를 의미하는 n 이 주어진다. ($1 \leq n \leq 5,000$) 마을은 1부터 n 까지 번호가 붙어있다. 지구이웨의 도로망은 다음 $n-1$ 개의 줄에 주어진다. 각 줄은 두개의 공백 하나로 구분된 정수 a, b 가 있으며, 마을 a 와 마을 b 사이에 직접 이어진 도로가 있다는 것을 의미한다. ($1 \leq a \leq b \leq n$)

출력 형식

첫째 줄에 가능한 호텔 배치의 가짓수를 정수 하나로 출력하여라.

서브태스크 1 (50 점)

- $n \leq 500$

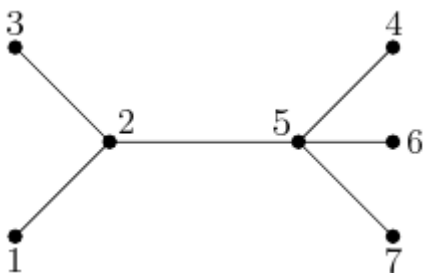
서브태스크 2 (50 점)

추가 제한조건이 없다.

예제

standard input	standard output
<pre> 7 1 2 5 7 2 5 2 3 5 6 4 5 </pre>	<pre> 5 </pre>

참고 사항



호텔을 세울 수 있는 위치는 다음과 같다: {1, 3, 5}, {2, 4, 6}, {2, 4, 7}, {2, 6, 7}, {4, 6, 7}

문제 3. 블록

입력 파일: standard input
출력 파일: standard output
시간 제한: 3 seconds
메모리 제한: 64 megabytes

어제 범수와 친구들은 유치원에서 다양한 색의 블록을 만들고 노느라 하루를 다 보냈다. 처음에 그들은 모형 빌딩을 만들었지만 곧 질려버렸다. 그래서 그들은 블록을 일렬로 배치하기로 했다. 단조로워 보이는 것을 막기 위해 같은 색의 두 블록을 옆에 놓지 않으려고 노력했다. 긴 시간 끝에 그들은 이 조건에 맞춰 블록을 놓는데에 성공했고, 부모님들은 다시 범수와 친구들을 집으로 데려갔다.

오늘 범수는 유치원에 일찍 왔다. 그는 어제 쌓은 블록이 계속 남아있다는 것에 만족했다. 하지만 그는 그 블록 위로 넘어지고 말았다. 범수는 그 블록들을 색별로 정리하고 다시 원래대로 복구하고 싶어 했다. 그가 기억한 것은 가장 왼쪽 끝과 오른쪽 끝의 블록 색 뿐이다.

범수에게 두 인접한 블록이 같은 색을 가지지 않고, 그가 기억하던 왼쪽 끝과 오른쪽 끝의 색을 유지하면서 어떻게 블록들을 배치해야 하는지 알려주자. 범수가 기억을 잘못하거나, 넘어지면서 블록들을 잃어버려서 새로 배치하는것이 불가능 할수도 있다.

입력 형식

첫째 줄에는 세 정수 블록 색의 가짓수 k , 가장 왼쪽 끝 블록의 색 p , 가장 오른쪽 끝 블록의 색 q 가 공백 하나로 구분되어 주어진다. ($1 \leq k \leq 1,000,000$, $1 \leq p, q \leq k$) 둘째 줄에는 k 개의 정수 i_1, i_2, \dots, i_k 가 공백 하나로 구분되어 주어진다. i_j 는 색 j 의 블록을 범수가 정확히 i_j 개 가지고 있다는 것을 의미한다. 총 블록의 갯수는 백만개를 넘지 않는다. 즉, $n = i_1 + i_2 + \dots + i_k \leq 1,000,000$ 이다.

출력 형식

첫째 줄에 n 개의 수를 공백 하나로 구분하여 출력하여라. 이 수는 가능한 블록 배치의 왼쪽 부터 오른쪽 까지의 색을 나타낸다. 만약 그런 배치가 없다면, 0을 출력하여라.

답이 여러개일 경우, 아무거나 출력하여도 좋다.

예제

standard input	standard output
3 3 1 2 3 3	3 2 1 3 2 3 2 1
3 3 1 2 4 2	0

참고 사항

첫번째 예제에서, 올바른 다른 정렬은 “3 1 2 3 2 3 2 1” 이다. 두번째 예제에서, 범수가 어딘가 실수를 해서 원래 조건을 만족하지 못했다.

문제 4. 택배

입력 파일: standard input
출력 파일: standard output
시간 제한: 3 seconds
메모리 제한: 64 megabytes

범수는 컴퓨터게임을 파는 ZIG회사에서 일한다. 이 회사는 ZIG회사에서 판매되는 게임들을 공급하는 다른 게임회사와 같이 일한다. 범수는 ZIG회사와 택배회사와의 협력도 조사하고 있다. 그는 각 게임 패키지를 공급한 택배회사가 어느 회사인지에 대한 기록을 가지고 있다. 범수는 어떠한 회사도 불공평한 기회를 가지지 않았으면 한다.

만약 어떤 택배회사가 특정한 기간에 절반 초과의 패키지를 보냈다고 하면, 그 회사가 그 기간을 독점했다고 말한다. 범수는 어떤 기간에 특정 택배회사가 독점을 한 적이 있는지 알아보고 싶다.

범수를 도와주자! 독점하는 택배회사가 있는지 없는지 결정하는 프로그램을 작성하여라.

입력 형식

첫째 줄에는 ZIG회사가 배송한 게임 패키지의 수를 나타내는 n 과, 게임 패키지를 배송한 날의 수를 나타내는 m 이 공백 하나로 구분되어 들어온다. ($1 \leq n, m \leq 500,000$) 택배 회사들은 1이상 최대 n 이하의 수로 표현된다. 둘째 줄에는 n 개의 정수 p_1, p_2, \dots, p_n 이 공백 하나로 구분되어 들어온다. ($1 \leq p_i \leq n$) p_i 는 i 번째로 배송된 게임 패키지의 번호를 나타낸다. 다음 m 개의 줄에는 쿼리가 한 줄에 하나씩 공백 하나로 구분된 두 수 a, b 로 주어진다. ($1 \leq a \leq b \leq n$) 이 쿼리는 a 번째 패키지부터 b 번째 패키지 까지 배송하는 기간에 독점하는 회사가 있는지 없는지 확인해야 한다. (이 기간은 a 번째 패키지와 b 번째 패키지를 포함한다.)

출력 형식

각 쿼리당 한 줄에 하나씩 답을 출력해야 한다. (총 m 줄을 출력해야 한다.) 각 줄에는 정수 하나가 있어야 한다. 쿼리에 해당하는 구간에 독점하는 회사가 있다면 그 회사의 번호를, 없다면 0을 출력 해야 한다.

서브태스크 1 (30 점)

- $n, m \leq 5,000$

서브태스크 2 (35 점)

- $n, m \leq 50,000$

서브태스크 3 (35 점)

추가 제한조건이 없다.

예제

standard input	standard output
7 5	1
1 1 3 2 3 4 3	0
1 3	3
1 4	0
3 7	4
1 7	
6 6	

문제 5. 뱀

입력 파일: standard input
출력 파일: standard output
시간 제한: 3 seconds
메모리 제한: 512 megabytes

뱀이 $3 \times n$ 격자판을 채우고 있다. 뱀은 1부터 $3n$ 까지 번호가 붙은 구간들로 이루어져 있다. 하나의 숫자는 정사각형 하나를 차지하고, 연속된 숫자 (1과 2, 2와 3, 등등등...)가 차지하는 정사각형은 뱀을 공유한다. 다음은 3×9 격자판을 채운 뱀의 예이다.:

7	6	5	4	17	18	19	20	21
8	1	2	3	16	15	26	25	22
9	10	11	12	13	14	27	24	23

뱀을 이루는 숫자들 몇개가 지워졌다. 원래 숫자를 복구하여라.

입력 형식

첫째 줄에는 격자판의 길이를 나타내는 정수 n 이 주어진다. ($1 \leq n \leq 1,000$) 다음 세개의 줄의 i 번째 줄은 공백 하나로 구분된 n 개의 정수가 주어진다. i 번째 줄의 j 번째 숫자를 a_{ij} 라고 할 때, $a_{ij} > 0$ 인 경우 뱀의 번호를 나타내고 $a_{ij} = 0$ 인 경우 뱀의 번호를 알 수 없다는 것을 의미한다. ($1 \leq a_{ij} \leq 3n$)

출력 형식

세개의 줄을 출력해야 하고, 각 줄은 n 개의 공백 하나로 구분된 정수여야 한다. $3n$ 개의 숫자는 1부터 $3n$ 까지의 수의 순열이어야 한다. 출력은 올바른 뱀이어야 한다. 즉, 입력 데이터와 모순되지 않아야 하고, 뱀의 조건을 만족해야 한다.

올바른 뱀의 배치가 존재함을 가정해도 된다. 답이 여러개인 경우, 아무것이나 하나를 출력한다.

서브태스크 1 (15 점)

- $n \leq 10$

서브태스크 2 (25 점)

- $n \leq 40$

서브태스크 3 (30 점)

- $n \leq 300$

서브태스크 4 (30 점)

추가 제한조건이 없다.

예제

standard input	standard output
9	7 6 5 4 17 18 19 20 21
0 0 5 0 17 0 0 0 21	8 1 2 3 16 15 26 25 22
8 0 0 3 16 0 0 25 0	9 10 11 12 13 14 27 24 23
0 0 0 0 0 0 0 0 23	

문제 6. 카드

입력 파일: standard input
출력 파일: standard output
시간 제한: 3 seconds
메모리 제한: 128 megabytes

n 개의 카드가 특정한 순서로 테이블에 놓여 있다. 각 카드마다 앞면과 뒷면에 수가 하나씩 총 두개 쓰여있다. 처음에, 모든 카드는 앞면이 위로 가게 놓여있다. 위대한 마법사 범수는 그의 특기인 이분탐색 카드 마법을 여러번 쓰려고 한다. 하지만 마법을 쓰기 위해서는 수열에 보이는 숫자가 비내림차순이어야 한다. 즉, 범수는 몇몇 카드들을 뒤집어서 뒷면이 보이게 만들 것이다.

그리고, 그의 마법은 청중들이 참가해야 한다. 아앗, 범수의 마법이 실패하기를 원하는 경쟁자가 몇몇의 청중을 배치해 놓았다. 현장에 입장할 때 마다 이런 (수상한) 청중은 빛의 속도로 두 카드 두 장을 바꿀 것이다. 카드를 바꾼 이후에 범수는 자기가 원하는 카드들의 앞뒷면을 바꿀 수 있지만, 그의 위대한 이분탐색 카드 마법을 실행하지 못할 수도 있다. 그 때는 모자에서 토끼를 꺼내는 시시한 마법을 보여줄 수 밖에 없다. 각 카드가 바뀐 이후에 범수가 그의 위대한 마법을 실행할 수 있는지의 여부를 구하는 프로그램을 작성하여라.

입력 형식

첫째 줄에는 카드의 갯수를 의미하는 정수 n 이 주어진다. ($2 \leq n \leq 200,000$) 다음 n 개의 줄에는 카드의 정보가 테이블에 놓여있는 순서대로 한 줄에 하나씩 주어진다. i 번째 줄에는 두 정수 x_i 와 y_i 가 공백 하나로 구분되어 들어온다. ($0 \leq x_i, y_i \leq 10^7$). 이 수들은 i 번째 카드에 적힌 숫자들을 나타낸다.: x_i 는 앞면, y_i 는 뒷면에 적힌 카드이다. 처음 카드 배치로 카드 마법을 사용하지 못할 수도 있다.

그 다음에 카드를 바꾸는 횟수 m 이 주어진다. ($1 \leq m \leq 1,000,000$) 다음 m 개의 줄은 카드 교환을 의미한다.: j 번째 줄은 a_j 와 b_j 가 공백 하나로 구분되어 주어진다. ($1 \leq a_j, b_j \leq n$) 이것은 j 번째 (수상한) 청중이 a_j 번째와 b_j 번째 카드를 바꾼다는 것을 의미한다.

출력 형식

프로그램은 m 개의 줄을 출력하여야 하고, 각 줄은 TAK이나 NIE여야 한다. 카드가 바뀐 이후에 범수가 카드 마술을 할 수 있으면 TAK, 아니면 NIE를 출력하여야 한다.

서브태스크 1 (30 점)

- $x_i = y_i$

서브태스크 2 (38 점)

- $n \leq 2,000$

서브태스크 3 (32 점)

추가 제한조건이 없다.

예제

standard input	standard output
4	2
2 5	3 4
3 4	1 3
6 3	
2 7	

문제 7. 범이자

입력 파일: standard input
출력 파일: standard output
시간 제한: 3 seconds
메모리 제한: 128 megabytes

지구이웨는 강을 낀 아름다운 마을이다. 강가에는 n 개의 집이 있고, 상류에서 하류까지 1부터 n 까지 번호가 붙은 집들이 있다. 지구이웨는 조용하고 모두가 행복한 마을이었다. 하지만 최근에 두 위험한 범이자인 범수와 상수가 마을에 나타났다. 그들은 이미 도둑질을 많이 해서 사람들은 집을 나가는것도 무서워 했다.

범수와 상수가 하는 도둑질은 단순하지 않고, 전체 집을 도둑질 한다. 하나의 집을 나오면, 서로를 향해 걸어가고 뒤로 돌아가지는 않는다. 범수는 하류쪽으로(숫자가 큰 쪽으로), 상수는 상류쪽으로(숫자가 작은 쪽으로) 걸어간다. 길을 걷는 동안, 서로가 만나기 전에는 각자가 몇몇 집으로 들어가서 물건들을 훔친다. 그 후에 그들은 집에서 만나서 그들이 훔친 물건들을 나눠갖는다. 지구이웨 주민들은 현기증이 났다. - 그들의 평화를 되찾아야 한다! 그래서 탐정 사과에게 도움을 요청했다.

형사는 도둑이 같은 색의 집에 살고 있지만 어떤 사람인지는 모른다고 판정했다. 방금 익명의 사람이 도둑들이 급습중이라고 밝혔다. 소식통은 자신의 안전을 두려워해서, 어떤 집으로 들어갈지 밝히지는 않았다. 하지만 그들의 색은 밝혔다. 또한, 도둑들은 한 종류의 색의 집을 한 번만 도둑질 할 것이라는 미신을 가지고 있다.

사과는 더 이상 기다릴 수 없다. 그는 도둑들이 만나는 장소에서 잠복할 것이다. 도둑들이 모이는 것이 가능한 장소를 모두 찾는 프로그램을 작성하여라.

입력 형식

첫째 줄에는 집의 수를 의미하는 정수 n 과 지구이웨에 있는 집의 색의 수를 의미하는 정수 k 가 공백 하나로 구분되어 주어진다. ($3 \leq n \leq 1,000,000$, $1 \leq k \leq 1,000,000$, $k \leq n$) 둘째 줄에는 n 개의 정수로 이루어진 수열 c_1, c_2, \dots, c_n ($1 \leq c_i \leq k$)가 공백 하나로 구분되어 주어진다. 이 수는 지구이웨에 있는 집들의 숫자들을 나열해 놓은 것이다.

셋째 줄에는 범수와 상수가 들어갈 집의 갯수를 나타내는 m, l 이 공백 하나로 구분되어 주어진다. ($1 \leq m, l \leq n$, $m + l \leq n - 1$) 넷째 줄에는 m 개의 정수 x_1, x_2, \dots, x_m 이 공백 하나로 구분되어 주어지고, 범수가 (자신의 집을 제외하고) 들어갈 집의 색을 차례로 나타낸다. ($1 \leq x_i \leq k$) 다섯째 줄에는 l 개의 정수 y_1, y_2, \dots, y_l 이 공백 하나로 구분되어 주어지고, 상수가 (자신의 집을 제외하고) 들어갈 집의 색을 차례로 나타낸다. ($1 \leq y_i \leq k$) $x_m = y_l$ 이고, 이 수는 범수와 상수가 자신이 도둑질한 물건들을 나눌 장소를 나타낸다. (그들은 그 집도 도둑질 할 것이다.)

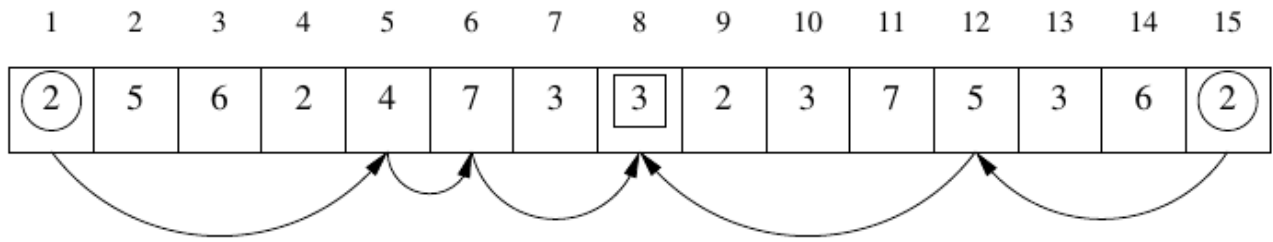
출력 형식

당신의 프로그램은 두개의 줄을 출력해야 한다. 첫째 줄은 위의 조건을 만족하면서 만날 수 있는 집의 갯수를 출력하여야 한다. 둘째 줄은 두 명이 만날 수 있는 집의 번호를 오름차순으로 출력해야 한다. 만약 상수와 범수가 만날 수 없다면, 첫째 줄은 0이어야하고, 둘째 줄은 비어있어야 한다.

예제

standard input	standard output
15 7	3
2 5 6 2 4 7 3 3 2 3 7 5 3 6 2	7 8 10
3 2	
4 7 3	
5 3	

참고 사항



위의 예제에서, 도둑들은 2번 색 (범수가 1번 혹은 4번 집, 상수가 15번 집) 혹은 6번 색 (범수가 3번 집, 상수가 14번 집)의 집에 살고 있을 것이다. 범수는 5번 집 (4번 색), 6번 집 (7번 색)을 차례로 들어가고 7번, 8번, 10번 집 (3번 색) 중의 하나에 들어갈 것이다. 상수는 12번 집 (5번 색)을 들어가고 7번, 8번, 10번 집 (3번 색) 중의 하나에 들어갈 것이다. 위의 그림은 범수가 1번집에 살고, 도둑들이 8번 집에서 모이는 상황을 나타낸다.

문제 8. 슈퍼컴퓨터

입력 파일: standard input
출력 파일: standard output
시간 제한: 3 seconds
메모리 제한: 128 megabytes

범수는 새로운 구조의 슈퍼컴퓨터를 개발했다. 이 슈퍼 컴퓨터는 같은 종류의 처리장치 여럿으로 이루어져 있다. 하나의 처리장치는 1 단위시간에 하나의 명령을 실행할 수 있다.

이 컴퓨터용 프로그램은, 순차적인 실행이 아니라 트리모양의 구조를 가지고 있다. 하나의 명령은 0개, 1개 또는 2개 이상의 후속명령을 가질 수 있다. a 가 b 의 후속명령이면, b 는 a 의 부모명령이다.

프로그램의 명령들은 병렬처리가 될 수 있다. 그리고, 다양한 순서로 실행될 수 있다. 단 하나의 제약조건은, 어떤 명령은 그 부모명령이 실행되기 전에는 실행될 수 없다는 것이다. 예를 들어, 한 명령의 후속 명령들은 어떠한 순서로도 병렬로 실행될 수 있다.

범수는 실행해야 할 어떤 프로그램이 있다. 범수는 그의 자원을 효율적으로 사용하는 것을 좋아하기 때문에, 처리장치의 갯수가 프로그램의 실행에 어떠한 영향을 미칠지 궁금해 졌다. 그는 프로그램과 처리장치의 갯수가 주어졌을 때, 슈퍼컴퓨터로 프로그램을 실행하는 최소시간을 구하고 싶어 한다.

입력 형식

첫째 줄에는 범수의 프로그램의 명령어 수를 의미하는 n 과 쿼리 갯수를 의미하는 q 가 공백 하나로 구분되어 주어진다. ($1 \leq n, q \leq 1,000,000$)

둘째 줄에는 q 개의 정수 k_1, k_2, \dots, k_q 가 공백 하나로 구분되어 주어진다. ($1 \leq k_i \leq 1,000,000$) k_i 는 범수의 i 번째 쿼리를 의미한다.

마지막 줄인 세번째 줄에는 $n - 1$ 개의 수열 a_2, a_3, \dots, a_n 이 공백 하나로 구분되어 주어진다. ($1 \leq a_i < i$) a_i 번 명령은 i 번 명령의 부모명령이다. 명령은 1부터 n 까지 번호가 붙어있고, 1번 명령은 프로그램의 가장 처음 명령이다.

출력 형식

프로그램은 q 개의 정수가 공백 하나로 구분된 한 줄을 출력해야 한다. i 번째 수는 k_i 개의 처리장치로 이루어진 슈퍼컴퓨터가 명령을 실행하는데 걸리는 최소시간을 출력해야 한다.

서브태스크 1 (20 점)

- $n \leq 1,000$
- $q \leq 10$

서브태스크 2 (15 점)

- $n \leq 30,000$
- $q \leq 50$

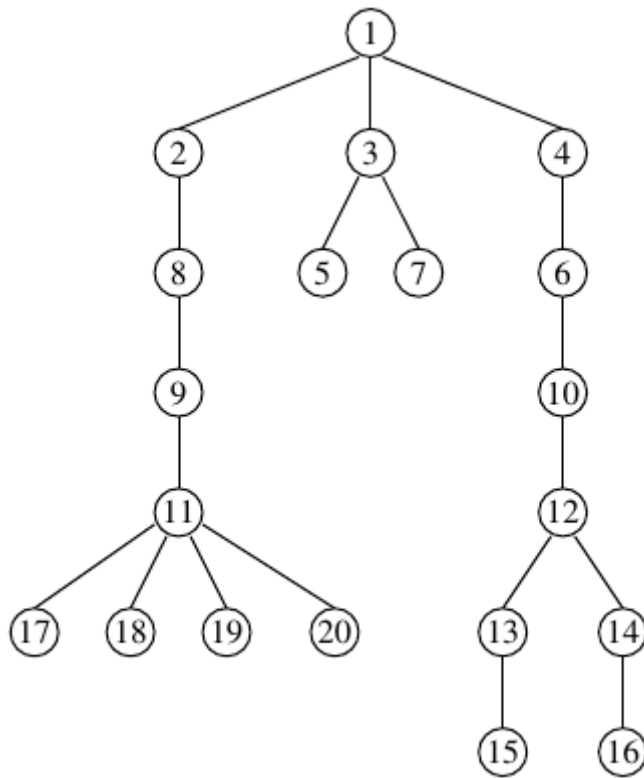
서브태스크 3 (65 점)

추가 제한조건이 없다.

예제

standard input	standard output
20 1 3 1 1 1 3 4 3 2 8 6 9 10 12 12 13 14 11 11 11 11	8

참고 사항



프로그램 명령은 다음과 같은 순서로 실행될 수 있다.

시간	명령			
1	1			
2	2	3	4	
3	5	6	7	
4	8	10		
5	9	12		
6	11	13	14	
7	15	16	17	
8	18	19	20	

문제 9. 아기새

입력 파일: standard input
출력 파일: standard output
시간 제한: 3 seconds
메모리 제한: 128 megabytes

지구이웨 국립공원에는 n 개의 나무가 일렬로 자라고 있다. 첫번째 나무의 꼭대기에는 마지막 나무의 꼭대기로 날고싶어하는 아기새가 있다. 아기새는 매우 작기 때문에 마지막 나무까지 한번에 날아갈 힘이 없을 수도 있다. i 번째 나무에 앉아있는 아기새는 한번 날기 시작해서 $i+1, i+2, \dots, i+k$ 번째 나무까지 날 수 있고, 그 후로 가려면 쉬어야 한다.

그리고 날아 올라가는 것은 날아 내려가는 것보다 힘든 일이다. 높이가, 시작한 나무의 높이보다 같거나 높은 곳으로 날아가면 피로가 1만큼 쌓인다. 그렇지 않은 경우에는 피로가 쌓이지 않는다.

아기새의 목표는 피로를 최대한 적게 쌓으면서 마지막 나무로 가는 것이다. 알다시피, 새들은 사회적인 동물이기 때문에 아기새는 똑같은 목표를 가지고 첫째 나무에서 마지막 나무로 가고싶어하는 서로 다른 체력 k 를 가진 아기새들이 있다. 아기새들을 도와주자!

입력 형식

입력의 첫째 줄에는 지구이웨 국립공원에 있는 나무의 수 n 이 주어진다. ($2 \leq n \leq 1,000,000$) 둘째 줄에는 n 개의 정수 d_1, d_2, \dots, d_n 이 공백 하나로 구분되어 주어진다. ($1 \leq d_i \leq 10^9$) d_i 는 i 번째 나무의 높이이다.

셋째 줄에는 마지막 나무로 가고 싶어하는 새의 수를 나타내는 q 가 주어진다. ($1 \leq q \leq 25$) 다음 q 개의 줄 중 i 번째 줄에는 i 번째 새의 힘을 나타내는 k_i 가 주어진다. ($1 \leq k \leq n-1$) 즉, i 번째 새는 쉬지 않고 연속으로 $k_i - 1$ 개의 나무를 지나칠 수 있다.

출력 형식

프로그램은 q 개의 줄을 출력해야 한다. i 번째 줄에는 i 번째 새가 마지막으로 가는데 까지 얻은 피로를 의미한다.

서브태스크 1 (33 점)

- $n \leq 1,000$

서브태스크 2 (44 점)

- $n \leq 100,000$

서브태스크 3 (23 점)

추가 제한 조건이 없다.

예제

standard input	standard output
9	2
4 6 3 6 3 7 2 6 5	1
2	
2	
5	

참고 사항

첫번째 새는 1, 3, 5, 7, 8, 9번 나무에서 쉬어간다. 3번째에서 5번째 나무로 갈 때, 7번째에서 8번째 나무로 갈 때 피로를 얻는다.

문제 10. 경주

입력 파일: standard input
출력 파일: standard output
시간 제한: 3 seconds
메모리 제한: 128 megabytes

지구이웨에서 매년 개최되는 자전거 경주가 올해에도 어김없이 개최된다. 지구이웨의 사이클리스트(자전거 타는 사람)들은 긴 거리를 다닌다. 모터사이클리스트(오토바이 타는 사람)들은 사이클리스트와 계속 마찰을 빚어왔고, 이 경주를 사보타주하기로 결심한다.

지구이웨는 n 개의 교차로가 있고, 단방향 도로로 연결되어있다. 신기하게도, 이 도로망에는 사이클이 없다. – 만약 교차로 u 에서 교차로 v 로 갈 수 있다면, 교차로 v 에서 교차로 u 로는 가지 못한다.

경주는 지구이웨에 도로를 따라 열릴것이다. 모터사이클리스트들은 그들의 불타오르는 바이크를 타고 대회가 열리는 교차로 중 하나에 가서 그 교차로를 막을 것이다. 전국 사이클리스트 협회는 당연히 다른 경로로 대회를 열 것이고, 이 경로가 매우 짧아서 사이클리스트들이 자신의 놀라운 지구력을 뽐낼 수 없을 수도 있다. 이는 모터사이클리스트들의 계획이다. 그들은 한 교차로를 막아서 최장거리를 제일 짧게 만들고 싶다.

입력 형식

입력의 첫째 줄에는 지구이웨의 교차로의 수와 도로의 수를 의미하는 n, m 이 공백 하나로 구분되어 주어진다. ($2 \leq n \leq 500,000, 1 \leq m \leq 1,000,000$) 교차로는 1번부터 n 번까지 번호가 붙어 있다. 다음 m 개의 줄의 i 번째 줄에는 두 정수 a_i, b_i 가 공백 하나로 구분되어 주어진다. ($1 \leq a_i, b_i \leq n, a_i \neq b_i$) 이것은 교차로 a_i 부터 b_i 까지 단방향 도로로 연결 되어 있다는 것을 의미한다.

출력 형식

프로그램은 첫째 줄에 두 개의 공백 하나로 구분된 정수를 출력하여야 한다. 첫번째 수는 모터사이클리스트들이 막아야 하는 교차로의 번호를 출력하고, 두번째 수는 사이클리스트들이 경주를 할 수 있는 가장 긴 경로의 도로의 수를 출력한다. 답이 여러개인 경우, 아무 답이나 출력해도 좋다.

서브태스크 1 (33 점)

- 모든 도로 i 에 대해, $a_i < b_i$.

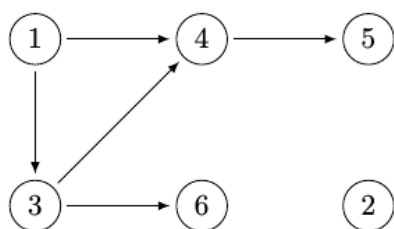
서브태스크 2 (67 점)

추가 제한 조건이 없다.

예제

standard input	standard output
6 5 1 3 1 4 3 6 3 4 4 5	1 2

참고 사항



문제 11. 범수크라프트

입력 파일: standard input
출력 파일: standard output
시간 제한: 3 seconds
메모리 제한: 128 megabytes

지구이웨 마을에는 n 개의 집이 $n - 1$ 개의 도로로 연결되어있다. 모든 두 집에 대해서, 하나로 부터 다른 하나로 가는 경로가 유일하다. 집은 1번부터 n 번까지의 번호가 붙어있다. 1번 집은 마을 이장 범수의 집이다. IoT(Internet of things)의 발달로 시골 마을에도 컴퓨터가 들어오게 되었고, n 대의 컴퓨터가 범수의 집에 도착했다. 각 집에 하나씩 컴퓨터를 나눠주는게 범수의 일이다. 지구이웨 마을 사람들은 이미 게임 범수크라프트를 하기로 약속한 상태이다.

범수는 자신의 트럭에 모든 컴퓨터를 담고 나누어줄 준비를 했다. 범수는 정확히 각 도로를 왕복할 연료만 가지고 있다. 각 집에 가서 범수는 컴퓨터 하나를 내려 두고 바로 다음 집으로 간다. 각 집은 컴퓨터를 받으면 바로 컴퓨터를 켜서 범수크라프트를 설치한다. 범수크라프트를 설치하는 시간은 얼마나 기술에 익숙하냐에 따라 다르다. 범수는 모든 집에 컴퓨터를 배달하고 집에 와서 바로 범수크라프트를 설치할 것이다. 도로 하나를 다니는데 걸리는 시간은 정확히 1분이고, (범수크라프트를 하려는 의지 덕분에) 컴퓨터를 트럭에서 내리는 시간은 무시할 수 있다.

범수를 도와서 범수를 포함한 모두가 최대한 빨리 범수크라프트를 같이 할 수 있게 도와주자. 즉, 모든 컴퓨터에 범수크라프트가 깔리는 데 필요한 최소 시간을 구하여라.

입력 형식

입력의 첫째 줄에는 지구이웨에 있는 집의 수 n 이 주어진다. ($2 \leq n \leq 1,000,000$) 둘째 줄에는 n 개의 정수 c_1, c_2, \dots, c_n 이 공백 하나로 구분되어 주어진다. ($1 \leq c_i \leq 10^9$) d_i 는 i 번째 집이 범수크라프트를 설치하는데 (분 단위로) 걸리는 시간이다.

다음 $n - 1$ 개의 줄에는 집을 잇는 도로를 설명한다. 각 줄은 두 정수 a, b 가 공백 하나로 구분되어 주어진다. ($1 \leq a < b \leq n$) 이것은 a 와 b 를 직접 잇는 도로가 존재한다는 의미이다.

출력 형식

프로그램은 하나의 수를 첫째 줄에 출력해야 한다. 이 수는 분 단위로 모두가 범수크라프트를 시작 할 수 있는 최소 시간이다.

서브태스크 1 (40 점)

- $n \leq 7,000$

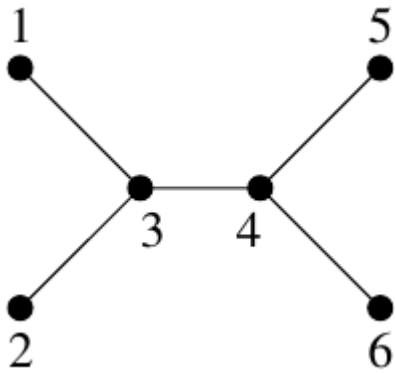
서브태스크 2 (60 점)

추가 제한 조건이 없다.

예제

standard input	standard output
6 1 8 9 6 3 2 1 3 2 3 3 4 4 5 4 6	11

참고 사항



범수는 컴퓨터를 다음 순서로 배달해야 한다.: 3, 2, 4, 5, 6, 1. 게임은 각각 11, 10, 10, 10, 8, 9분 후에 끝날 것이다. 그러므로 모두가 11분 후면 게임을 플레이 할 수 있다.

만약 범수가 3, 4, 5, 6, 2, 1 순서로 게임을 배달 했다면, 게임은 11, 16, 10, 8, 6 분 후에 끝날 것이다. 그래서 모두가 16분 후에 게임을 플레이 해야 한다.

문제 12. 세계일주

입력 파일: standard input
출력 파일: standard output
시간 제한: 3 seconds
메모리 제한: 24 megabytes

몇년간의 노력 끝에, 범수는 결국 조종사면허를 땀다. 이 사실을 기념하기 위해 그는 항공기를 사서 자신이 사는 행성인 **3-셋별** 근처를 비행하려고 한다. 구체적으로, 범수는 적도 근처를 한바퀴 비행하려고 한다. 불행하게도, 행성이 매우 크기 때문에 중간에 연료를 재충전하는것이 필요하다. 각 항공기가 연료가 꽉 찬 상태에서 얼마나 멀리 비행할 수 있는지가 알려져 있다. 적도 근처에는 많은 수의 공항이 있고 항공기가 공항에 착륙하면 연료를 충전할 수 있다. 항공기를 사는것은 중요한 일이길 때문에, 범수는 당신의 도움을 요청했다. 그는 자신이 구매를 생각하는 다양한 항공기 종류를 알려줄 것이다. 각 항공기들은 한번에 얼마나 멀리 비행할 수 있는지가 다르다. 각 항공기 모델에 따라 여행을 하기 위한 최소한의 (마지막 착륙을 포함한) 착륙 횟수가 궁금해 졌다. 각 항공기 모델에 따라, 여행은 자신이 원하는 각자 다른 공항에서 시작할 수도 있다.

입력 형식

입력의 첫째 줄에는 적도에 있는 공항의 수 n 과 범수가 구매를 생각하는 항공기 종류의 수 s 가 공백 하나로 구분되어 주어진다. ($2 \leq n \leq 1,000,000$, $1 \leq s \leq 100$).

둘째 줄에는 공항 간의 간격을 의미하는 n 개의 양의 정수 l_1, l_2, \dots, l_n 이 공백 하나로 구분되어 주어진다. ($l_1 + l_2 + \dots + l_n \leq 10^9$) l_i 는 i 번째 공항과 $i+1$ 번째 공항($i = n$ 인 경우는 n 번째 공항과 첫번째 공항)의 거리를 km단위로 의미한다.

셋째 줄에는 s 개의 정수 d_1, d_2, \dots, d_s ($1 \leq d_i \leq l_1 + l_2 + \dots + l_n$)가 공백 하나로 구분되어 주어진다. d_i 는 착륙 하지 않고 비행기가 날 수 있는 거리를 km단위로 의미한다.

출력 형식

프로그램은 s 개의 줄을 출력해야 한다. i 번째 줄에는 3-셋별을 적도를 따라 한바퀴 돌기 위한 최소한의 이착륙 횟수를 나타내는 정수 하나를 출력하거나, 비행이 불가능 하면 NIE를 출력해야 한다.

서브태스크 1 (20 점)

- $n \leq 1,000$

서브태스크 2 (30 점)

- $n \leq 100,000$

서브태스크 3 (18 점)

- $s \leq 5$

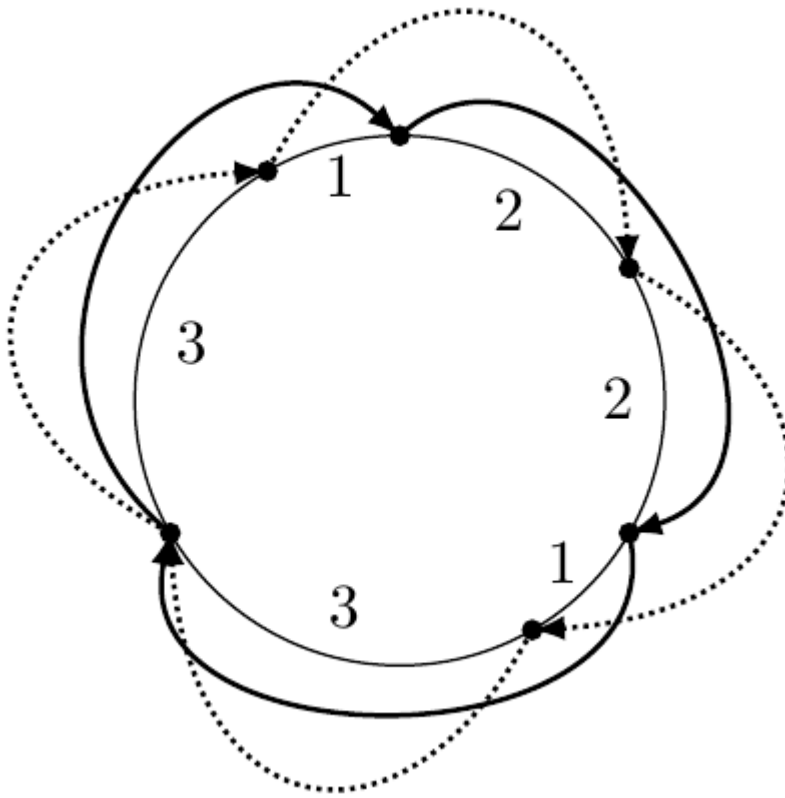
서브태스크 4 (32 점)

추가 제한 조건이 없다.

예제

standard input	standard output
6 4	4
2 2 1 3 3 1	NIE
3 2 4 11	3
	2

참고 사항



실선은 착륙하지 않고 비행기가 날 수 있는 거리가 4km일 때의 최적의 비행을 보여준다. 점선은 3km 일 때 이다.

문제 13. 개미집

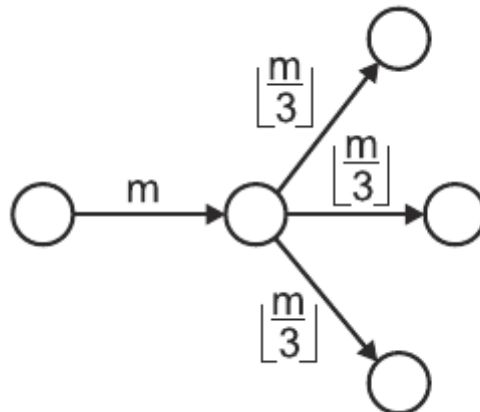
입력 파일: standard input
출력 파일: standard output
시간 제한: 3 seconds
메모리 제한: 128 megabytes

개미들은 음식을 찾아 버려진 개미 언덕을 찾아 해매고 있다. 개미 언덕은 n 개의 방과 방을 연결하는 $n - 1$ 개의 통로로 이루어진다. 모든 방은 모든 다른 방으로의 통로로 이루어진 유일한 경로가 있다. 즉 방과 통로는 트리구조이다.

개미 언덕의 통로 하나와만 연결된 모든 방에는 입구가 있다. 입구마다 각각 개미 m_1, m_2, \dots, m_g 마리로 이루어진 g 개의 개미 그룹들이 있다. 이 그룹들은 각각 개미 언덕에 들어가고, 개미 언덕 안에 개미가 없어지면 다른 그룹이 들어가는 것을 반복한다. 개미 언덕 안에서, 개미는 다음과 같은 방법으로 탐험을 한다.:

- 개미 그룹이 한번도 탐험하지 않은 통로가 d 개가 있는 방으로 들어가면, 개미 그룹은 각 통로에 d 개의 크기가 같은 그룹으로 나뉘어진다. $d = 0$ 인 경우에는, 개미 그룹은 언덕을 나간다.
- 개미들이 같은 그룹으로 나누어 질 수 없는 크기라면, 힘이 센 개미가 약한 개미를 먹는다. 개미 그룹의 크기가 0이 될 수 있으니 항상 나누어 질 수 있다. 개미들이 같은 크기로 나누어지기 위해서는 자기 자신을 먹기도 한다.

다음 그림은 m 마리의 개미가 세 개의 탐험하지 않은 통로가 있는 방으로 들어가, $\lfloor \frac{m}{3} \rfloor$ 마리로 나뉘는 그림이다.



배고픈 개미핥기는 통로중 하나를 파고 들어가서 개미들을 먹으려고 한다. 하지만 개미들 처럼 개미핥기도 숫자들에 대해 간간하기 때문에 그 통로를 지나는 개미가 정확히 k 마리 일때만 개미들을 먹는다. 개미핥기가 몇 마리의 개미를 먹는지 구해주자.

입력 형식

첫째 줄에는 세 개의 정수 n, g, k 가 공백 하나로 구분되어 주어진다. ($2 \leq n, g \leq 1,000,000, 1 \leq k \leq 10^9$) 이 숫자들은 각각 방의 수, 개미 그룹의 수, 개미핥기가 한번에 먹는 개미 수를 의미한다. 방은 1번부터 n 번까지의 번호가 붙어있다.

둘째 줄에는 g 개의 수 m_1, m_2, \dots, m_g ($1 \leq m_i \leq 10^9$)가 공백 하나로 구분되어 주어진다. m_i 는 i 번째 그룹의 개미 수를 나타낸다. 다음 $n - 1$ 개의 줄은 개미 언덕의 통로에 대한 정보가 주어진다. $n - 1$ 개의 줄 중 i 번째 줄은 두 정수 a_i 와 b_i 이 공백 하나로 구분되어 주어진다. ($1 \leq a_i, b_i \leq n$) 이것은 방 a_i 와 방 b_i 가 통로를 통해 이어져있다는 것을 의미한다. 개미핥기는 첫번째로 나타나는 통로를 파고 들어간다.

출력 형식

첫째 줄에 개미핥기가 먹은 개미의 수를 나타내는 정수 하나를 출력하여라.

서브태스크 1 (20 점)

- $n \leq 100$
- $g \leq 100$

서브태스크 2 (50 점)

- 언덕에 들어가는 개미 그룹의 수는 1,000,000마리를 넘지 않는다.

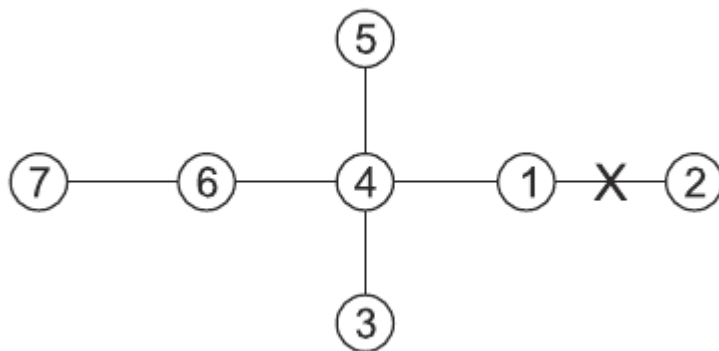
서브태스크 3 (30 점)

추가 제한조건이 없다.

예제

standard input	standard output
7 5 3 3 4 1 9 11 1 2 1 4 4 3 4 5 4 6 6 7	21

참고 사항



각 입구는 방 2, 3, 5, 7 옆에 있고, 통로 옆에는 5 그룹의 개미들이 있다. 개미핥기는 2번 방에서 시작하는 첫번째 그룹의 개미들과, 3번, 5번, 7번 방에서 시작하는 4번째, 5번째 그룹의 개미들을 먹을 것이다.

문제 14. 관광

입력 파일: standard input
출력 파일: standard output
시간 제한: 3 seconds
메모리 제한: 256 megabytes

아름다운 자연관광을 가진 지구이웨의 왕 범수는, 지구이웨가 많은 관광객들을 유치해서 돈을 써 국고에 도움이 되어야 한다고 생각한다. 하지만 현실은 그의 꿈과는 멀다. 그래서 왕은 의원에게 문제를 조사하라고 하였다. 의원은 도로망이 제대로 갖춰있지 않아서 외국인들이 싫어한다는 사실을 알았다.

지구이웨에는 n 개의 마을이 있고, m 개의 양방향 도로가 있으며, 도로가 서로 다른 두 마을을 잇는다. 도로들은 터널이나 고가도로가 있을 수 있다. 한 마을에서 도로를 통해서 다른 마을로 갈 수 있다는 보장은 없다.

의원은 현재의 도로망이 긴 여행에 적합하지 않다는 것을 알 수 있다. 어떤 사람이 여행을 시작하면, 같은 도시를 두번 지나지 않고서는 10개의 도시보다 많은 도시를 지날 수 없다.

국고가 제한되어 있기 때문에, 새로운 도로를 지을 수는 없다. 하지만, 범수는 짧은 여행이 가능하다는것에 알리는 관광정보지점(Tourist Information Point, TIP)를 만들기로 했다. 모든 마을 마다, 그 마을 혹은 그 마을과 바로 인접한 마을에 TIP가 있어야 한다. 그리고, 어떤 마을에 TIP를 만드는 비용은 모든 마을마다 알려져 있다. 왕을 도와서 위의 조건을 만족하는 가장 저렴한게 TIP를 짓는 방법을 구하여라.

입력 형식

첫째 줄에는 지구이웨의 마을 수를 나타내는 n 과 도로 수를 나타내는 m 이 공백 하나로 구분되어 주어진다. ($2 \leq n \leq 20,000$, $0 \leq m \leq 25,000$) 마을은 1번부터 n 번까지의 번호가 붙어있다. 입력의 둘째 줄은 n 개의 정수 c_1, c_2, \dots, c_n 으로 이루어져 있고, c_i 는 i 번째 마을에 TIP를 짓는 비용을 나타낸다.

그리고 지구이웨의 도로망을 나타내는 정보가 주어진다. 다음 m 개의 줄에는 두 정수 a_i, b_i ($1 \leq a_i < b_i \leq n$) 이 공백 하나로 구분되어 주어진다. 이것은 마을 a_i 와 b_i 가 도로로 연결되어있다는 것을 의미한다. 모든 두 마을 사이에는 최대 하나의 도로가 있다.

출력 형식

첫째 줄에 TIP를 짓는 최소 비용을 정수 하나로 출력해야 한다.

서브태스크 1 (20 점)

- $n \leq 20$

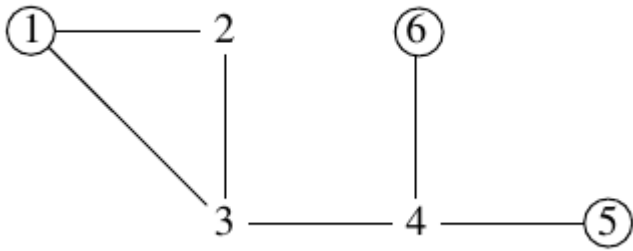
서브태스크 2 (80 점)

추가 제한조건이 없다.

예제

standard input	standard output
6 6 3 8 5 6 2 2 1 2 2 3 1 3 3 4 4 5 4 6	7

참고 사항



가장 저렴하게 TIP를 짓기 위해서는 1번, 5번, 6번 마을에 지어야 한다. ($3+2+2=7$ 만큼의 돈이 든다.)

문제 15. 태양광 램프

입력 파일: standard input
출력 파일: standard output
시간 제한: 3 seconds
메모리 제한: 128 megabytes

범수는 크고 아름다운 정원을 가지고 있다. 그는 밤에도 아름다운 광경을 즐기기 위해서 정원에 램프를 설치했다.

이 램프는 방향이 있다. 즉, 이 모든 램프들은 정해진 특정한 각도만 비출 수 있다. 게다가, 범수는 같은 방향을 비추도록 배치를 해 놓았다. 그리고 이 램프는 태양광 램프다. 이상하게도, 배터리가 없는 태양광 패널로 되어있다. 당신은 이 램프가 밤에 비추어야 해서 쓸모 없다고 생각할 수 있지만, 다른 태양광 램프가 이 램프를 비춘다면, 이 램프는 빛을 낼 수 있다.

지금부터 범수는 전기를 공급해서 램프를 키려고 한다. 문제를 간단히 하기 위해, 우리는 램프를 1번부터 n 번까지 번호를 붙인다. 즉, i 번 램프는 시간 i 에 전기가 공급된다. 범수(와 당신)에게 남은 일은, 각 램프가 언제 빛을 비추기 시작하는가이다. 범수에게 이 문제에 대한 답을 알려주는 프로그램을 작성하여라.

입력 형식

입력의 첫째 줄에는 범수가 설치한 램프의 수를 나타내는 정수 n 이 주어진다. ($1 \leq n \leq 200,000$) 둘째 줄에는 빛을 비추는 영역을 의미하는 네 개의 정수 X_1, Y_1, X_2, Y_2 가 공백 하나로 구분되어 주어진다. ($-10^9 \leq X_i, Y_i \leq 10^9, (X_i, Y_i) \neq (0, 0)$) 램프가 (x, y) 에 배치되어 있다면, (x, y) 에서 두개의 빛 $(x + X_i, y + Y_i)$ 이 만드는 각 중 작은 각도에 램프를 비춘다. (경계를 포함하고, 각은 항상 180도 보다 작다. 범수가 실수로 레이저를 사버려서 각이 0도일 수도 있다.)

다음 n 개의 줄은 i 번 램프가 (x_i, y_i) 에 위치해 있다는 것을 나타내는 두 정수 x_i 와 y_i 가 공백 하나로 구분되어 주어진다. ($-10^9 \leq x_i, y_i \leq 10^9$) 어떤 두 램프도 같은 위치에 있지 않다는 것이 보장된다.

다음 줄은 n 개의 정수 k_1, k_2, \dots, k_n 이 공백 하나로 구분되어 주어진다. 이 수는 i 번째 램프가 k_i 개의 램프의 빛을 받으면 빛을 내기 시작할 것을 의미한다.

출력 형식

프로그램은 첫째 줄에 n 개의 정수 t_1, t_2, \dots, t_n 을 공백 하나로 구분해서 출력해야 한다. 숫자 t_i 는 i 번 램프가 시간 t_i 에 빛을 내기 시작할 것을 의미한다.

서브태스크 1 (30 점)

- $n \leq 1000$

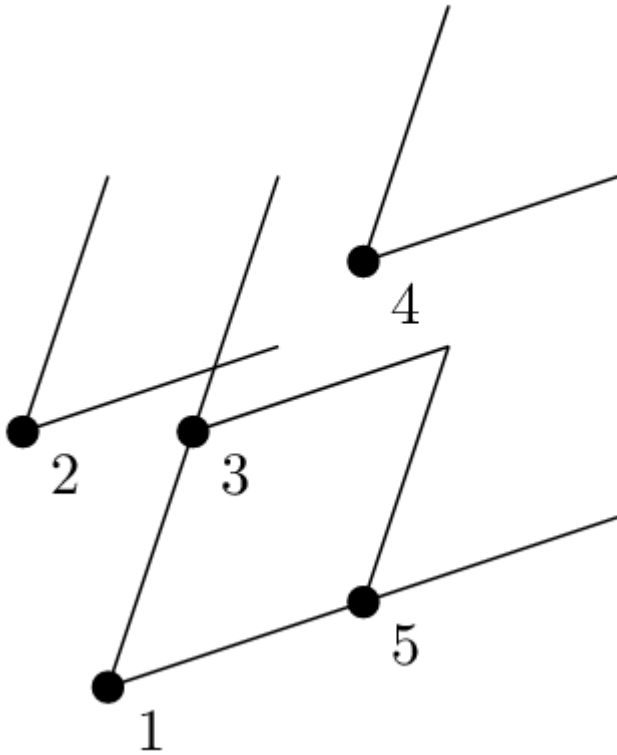
서브태스크 2 (70 점)

추가 제한조건이 없다.

예제

standard input	standard output
5 3 1 1 3 2 1 1 4 3 4 5 6 5 2 1 2 1 3 2	1 2 1 2 5

참고 사항



시간 1에 1번 램프를 켜면, 3번 램프도 같이 불이 켜진다. 2번 램프가 켜지면, 4번 램프도 같이 켜진다. (1, 2, 3번 램프가 빛을 비춘다.)

문제 16. 태양광 패널

입력 파일: standard input
출력 파일: standard output
시간 제한: 3 seconds
메모리 제한: 128 megabytes

신재생에너지를 개발하기로 결심한 범수는 태양광 패널 공장을 시작했다. 마치 금광을 찾은 것 처럼 며칠 동안 n 명의 고객이 문을 두드렸다. 각 고객은 너비와 높이가 정해진 직사각형 패널을 주문했다.

패널은 정사각형 모양의 광전지로 이루어져 있다. 이 전지의 한 변은 모든 단위 길이의 정수 크기로 제작될 수 있지만, 하나의 패널 안에 있는 모든 전지는 크기가 같아야 한다. 생산 규모가 클수록 경제성이 좋기 때문에, 전지가 클수록 효율적이다. 그래서 범수는 주문한 각 패널에 대해 최대 크기의 전지의 한 변의 길이를 알고 싶어 했다.

입력 형식

입력에 첫째 줄에는 주문을 받은 정수 n 이 주어진다. ($1 \leq n \leq 1,000$) 다음 n 개의 줄은 패널에 대한 정보를 나타낸다. 각 줄은 s_{min} , s_{max} , w_{min} , w_{max} 가 공백 하나로 구분되어 주어지며, 이 수는 각각 주문한 패널에 요구된 최소, 최대 너비와 최소, 최대 높이를 각각 나타낸다. ($1 \leq s_{min} \leq s_{max} \leq 10^9$, $1 \leq w_{min} \leq w_{max} \leq 10^9$)

출력 형식

프로그램은 정확히 n 개의 줄을 출력해야 한다. i 번째 줄은 i 번째 패널을 만들 때 쓸 수 있는 최대 크기의 전지의 한 변의 길이를 구해야 한다.

서브태스크 1 (20 점)

- $s_{max}, w_{max} \leq 10^7$
- $n \leq 10$

서브태스크 2 (55 점)

- $s_{max}, w_{max} \leq 10^7$

서브태스크 3 (25 점)

추가 제한조건이 없다.

예제

standard input	standard output
4	8
3 9 8 8	7
1 10 11 15	2
4 7 22 23	5
2 5 19 24	

참고 사항

범수는 8×8 (전지 하나), 7×14 (전지 두개), 4×22 혹은 6×22 (전지 각 22개 혹은 33개), 5×20 (전지 네개) 크기의 태양광 패널 네 개를 생산할 것이다.

문제 17. 화물

입력 파일: standard input
출력 파일: standard output
시간 제한: 3 seconds
메모리 제한: 64 megabytes

지구이웨와 카제룬 마을의 지하철역은 트랙 하나가 있는 철길로 연결되어 있다. 한 마을에서 다른 마을로 가는 데에는 s 분이 든다. 하지만 역을 출발하는 열차는 최소한 1분의 간격을 두어야 한다. 그리고 어떤 경우라도 철길에 있는 모든 열차들은 같은 방향으로 가고 있어야 한다.

계획된 시간표에 따르면 카제룬에 도착한 n 개의 화물차가 지구이웨로 가야한다. 그 열차는 카제룬에서 실은 물건을 가지고 달리고 있고, 지구이웨로 돌아가야 한다. 문제를 단순히 하기 위해, 우리는 물건을 싣는 데에 시간이 걸리지 않는다고 가정한다.

지구이웨로 돌아가는 마지막 열차의 시간을 최소화 하자.

입력 형식

입력에 첫째 줄에는 두 정수 n, s 가 공백 하나로 구분되어 들어온다. ($1 \leq n \leq 1,000,000, 1 \leq s \leq 10^9$) 둘째 줄에는 n 개의 정수 t_1, t_2, \dots, t_n 이 공백 하나로 구분되어 주어진다. ($0 \leq t_1 \leq t_2 \leq \dots \leq t_n \leq 10^9$). 이 숫자는 지구이웨에 도착하는 열차의 시간을 차례대로 나타낸다.

출력 형식

프로그램은 첫째 줄에, 지구이웨로 돌아가는 마지막 열차의 도착 시간을 최소화 한 값을 출력하여야 한다.

서브태스크 1 (25 점)

- $n \leq 400$

서브태스크 2 (25 점)

- $n \leq 5000$

서브태스크 3 (50 점)

추가 제한조건이 없다.

예제

standard input	standard output
3 4 1 8 11	20

참고 사항

시간을 최소화 하기 위해서, 열차는 지구이웨에서 1, 9, 11분에 출발하고, 카제룬에서는 5, 15, 16분에 출발해야 한다.