

ucpc 2020

Finals

Official Problemset

전국 대학생 프로그래밍 대회 동아리 연합
여름 대회 2020 본선

주최 전국 대학생 프로그래밍 대회 동아리 연합

주관  고려대학교 SW중심대학사업단

후원  STARTLINK

NAVER 

MINDs Lab.

ALGO SPOT

 K KENNYSOFT

SOLVED. 

문제 목록

문제지에 있는 문제가 총 12문제가 맞는지 확인하시기 바랍니다.

- A 전단지 돌리기
- B 던전 지도
- C 함수 복원
- D 소가 길을 건너간 이유 2020
- E 지도 설치
- F 애완 트리
- G 그건 망고가 아니라 고양이에요
- H 레이저 연구소
- I 빛의 전사 크리퓨어
- J 관광 사업
- K 데이터 제작
- L 피자 배틀

모든 문제의 메모리 제한은 1GB로 동일합니다.

문제 A. 전단지 돌리기

시간 제한 1 초
메모리 제한 1024 MB

현민이는 트리 모양의 길 위에서 오토바이를 타고 전단지를 돌리려고 한다. 현민이의 목표는 케니소프트에서 출발하여 모든 노드에 전단지를 돌리고, 다시 케니소프트로 돌아오는 것이다. 현민이는 힘이 좋기 때문에 현재 노드에서 거리가 D 이하인 모든 노드에 전단지를 돌릴 수 있다.

날씨가 매우 덥기 때문에, 현민이는 최소한만 이동해서 목표를 달성하고 싶다! 현민이를 위해 현민이가 이동해야 하는 총 거리를 구해주자.

입력

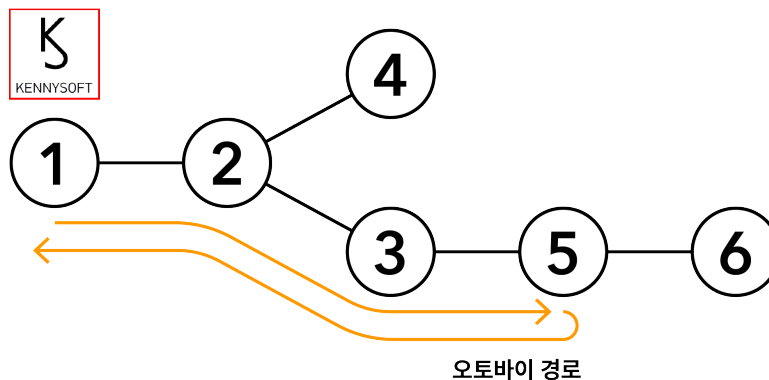
첫번째 줄에는 노드의 개수 N ($1 \leq N \leq 100\,000$)과 케니소프트의 위치 S ($1 \leq S \leq N$), 힘 D ($0 \leq D \leq N$)이 주어진다. 두 번째 줄부터 N 번째 줄까지, 트리의 간선 정보를 의미하는 두 자연수 x, y 가 공백으로 구분되어 주어진다. 이는 x 번 노드와 y 번 노드가 연결되어 있음을 의미한다. ($1 \leq x, y \leq N, x \neq y$)
주어지는 연결관계는 트리를 구성하며, 모든 간선의 길이는 1이다.

출력

현민이가 목표를 완수하기 위해 이동해야 하는 최소 거리를 출력하여라.

입출력 예시

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



이 페이지는 공백입니다

문제 B. 던전 지도

시간 제한 1 초
메모리 제한 1024 MB

동현이는 로그라이크 던전 탐험 게임을 개발하고 있다. 이 게임에서 던전은 N 행 M 열의 격자 형태로 방이 놓여 있는 구조이다. 아래쪽에서 i 번째 행의 왼쪽에서 j 번째에 있는 방을 (i, j) 라고 하자.

던전의 각 방에는 **R** 또는 **U**가 적혀 있다. 플레이어는 던전의 임의의 방에서부터 시작해 방을 하나씩 꺼 나가는데, 이번에 깬 방이 (i, j) 일 때 그 방에 **R**이 적혀 있었다면 $(i, j+1)$ 로 이동하고, **U**가 적혀 있었다면 $(i+1, j)$ 로 이동한다. 만약 새로 이동할 좌표에 해당하는 방이 없다면 던전을 탈출하게 된다.

동현이는 던전 지도를 효율적으로 만들기 위해 다음과 같은 방식을 사용하였다.

- 지도의 한 행을 구성하는 K 종류의 블록을 미리 만들어 놓고, 각 블록에 알파벳 대문자의 처음 K 글자 중 하나를 대응시킨다. 각 블록은 **R**과 **U**로만 이루어진 길이 M 의 문자열이다.
- 알파벳 대문자의 처음 K 종류로 이루어진 길이 N 의 문자열을 하나 생성한 뒤, 그 문자열의 i 번째 글자에 대응되는 블록으로 던전의 아래에서 i 번째 행을 만든다.

던전의 가장 오른쪽 위, 즉 (N, M) 은 보스 방으로, 던전에서 가장 어려운 방인 대신 깨면 막대한 보상이 주어진다. 시작 위치에 따라 보스 방에 도달 가능할 수도 있고 없을 수도 있는데, 동현이는 지금 만든 지도에서 과연 보스 방에 도달할 수 있는 시작 방이 몇 개나 되는지 궁금해졌다.

하지만, 동현이는 지도 생성 말고도 할 일이 많다. 여러분이 동현이를 대신해 의문점을 해결해 주자!

입력

첫 번째 줄에 던전의 행 개수 N , 열 개수 M , 블록의 종류 K 가 공백으로 구분되어 주어진다. ($1 \leq N, M \leq 200\,000$, $1 \leq K \leq 26$)

두 번째 줄부터 K 개의 줄에 **R**과 **U**로만 이루어진 길이 M 의 문자열이 주어진다. 이 K 개의 줄 중 i 번째 줄은 i 번째 알파벳 대문자에 대응되는 블록이다.

그 다음 줄에 알파벳 대문자의 처음 K 종류로만 이루어진 길이 N 의 문자열이 주어진다. 이 문자열의 j 번째 글자는 던전의 아래에서 j 번째 행을 구성하는 블록을 의미한다.

출력

첫 번째 줄에 주어진 지도에서 (N, M) 에 도달할 수 있는 시작 방의 개수를 출력한다. (N, M) 도 시작 방이 될 수 있음에 유의하라.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
3 4 3 RURU RRUR UURU CBA	7
5 3 1 RRU AAAAA	15

이 페이지는 공백입니다

문제 C. 함수 복원

시간 제한 1.5 초
메모리 제한 1024 MB

자연수 N 이 주어질 때, 1 이상 N 이하의 자연수 i 에 대해 정의되는 함수 f 가 있다. 모든 i 에 대해 $f(i)$ 또한 1 이상 N 이하의 자연수다.

f 에 대한 함수 그래프를 정점 i 에서 정점 $f(i)$ 로 향하는 단방향 간선들로 이루어진 그래프라 부르자. 이 그래프는 f 에 따라 유일하게 결정됨을 알 수 있다.

우리는 함수 f 를 알지 못하지만, f 에 대한 함수 그래프에서 모든 정점에 대한 도달 가능성 정보를 가지고 있다. 정점 u 에서 정점 v 에 도달 가능하다는 것은 0개 이상의 간선을 통해 정점 u 에서 정점 v 로 갈 수 있다는 뜻이다.

함수 f 로 가능한 경우의 수를 구하여라.

입력

첫 줄에 N 이 주어진다. ($1 \leq N \leq 500$)

그 후, N 개의 줄에 걸쳐 f 에 대한 함수 그래프에서 도달 가능성 정보가 공백으로 구분되어 주어진다. i 번째 줄의 j 번째 수는 i 에서 j 에 도달 가능하면 1, 그렇지 않으면 0이다. 가능한 함수 f 가 존재하지 않는 경우는 주어지지 않는다.

출력

함수 f 로 가능한 경우의 수를 $10^9 + 7$ 로 나눈 나머지를 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
6 1 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 0 0 1 1 1 1 1 0 0 0 0 0 0 1	6
9 1 0 1 1 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 1 1 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 1 1 1 1 1 0 0 1 0 0 0 0 0 0 1 1 0 1 1 0 0 0 0 1 0 0 0 1	6

이 페이지는 공백입니다

문제 D. 소가 길을 건너간 이유 2020

시간 제한 1 초
메모리 제한 1024 MB

농부 존의 목장에는 y 좌표가 0보다 크고 L 보다 작은 영역에 커다란 길이 나왔다. 소들은 길가에 있는 헛간에서 살고 있으며 서로 다른 곳에 살고 있다. 길의 위쪽에 살고 있는 소들은 총 N 마리이며 i 번째 소는 (U_i, L) 에 살고 있다. 길의 아래쪽에 살고 있는 소들은 총 M 마리이며 j 번째 소는 $(D_j, 0)$ 에 살고 있다.

소들이 자꾸 길을 건너는 게 못마땅한 농부 존은 길에 구덩이를 파서 소들이 길을 건너지 못하게 하였다. 길을 자주 건너던 소들은 다들 항의했지만 소들의 항의는 받아들여지지 않았다.

그러던 어느 날, 소들은 꿀벌에게 쏘이면 잠깐 동안 하늘을 날 수 있다는 사실을 깨달았다. 이 사실이 퍼지자 소들은 다시 자유롭게 길을 건너기 시작했다. 소들은 하늘을 날아서 길을 건너는 게 너무 좋은지 굳이 길을 건너지 않아도 될 상황에서도 길을 여러 번 건넜다.

하늘을 나는 동안에는 소들이 방향을 조절할 수 없기 때문에 소들이 공중에서 부딪히는 사고가 종종 발생했다. 사고가 나지 않기 위해 소들의 리더 베시는 아래 조건을 만족하도록 항로를 만들어서 사고가 일어나지 않도록 하려고 한다.

1. 항로는 위쪽 헛간과 아래쪽 헛간을 잇는 선분이다.
2. 항로는 $N + M - 1$ 개만 만든다.
3. 모든 헛간들이 항로를 통해서 연결되어야 한다.
4. 임의의 두 항로는 헛간이 아닌 곳에서 교차하면 안 된다.

소들은 한번 날 때 비행 거리의 제곱만큼 힘이 든다. 만약 소가 여러 번 난다면 비행 거리의 제곱의 합만큼 힘이 든다. 이에 따라 베시는 두 헛간 사이의 거리를 '항로만 이용하여 한 헛간에서 다른 헛간까지 갈 때 필요한 최소 힘'으로 정의했다.

베시는 모든 헛간 쌍에 대한 거리의 합을 최소화하려고 한다. 베시를 도와 최적의 항로를 구하는 프로그램을 작성하라.

입력

첫 번째 줄에 위쪽에 살고 있는 소들의 수 N , 아래쪽에 살고 있는 소들의 수 M , 길의 폭 L 이 주어진다. ($1 \leq N, M \leq 3\,000, 1 \leq L \leq 30\,000$)

두 번째 줄에는 위쪽에 살고 있는 소들의 x 좌표가 오름차순으로 주어진다.

세 번째 줄에는 아래쪽에 살고 있는 소들의 x 좌표가 오름차순으로 주어진다.

모든 좌표는 0 이상 30 000 이하의 정수이다.

출력

베시가 최적의 항로를 구축했을 때, 모든 헛간 쌍에 대해 두 헛간의 거리들의 합을 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
3 2 1 1 3 5 2 4	40

이 페이지는 공백입니다

문제 E. 지도 설치

시간 제한 2 초
메모리 제한 1024 MB

듀벤은 강아지 산책로 사용권들의 편의를 위해서 산책로에 지도들을 설치하려고 한다. 산책로는 여러 개의 정점과 단방향 간선으로 이루어진 그래프로 생각할 수 있고, 서로 다른 출발점과 도착점이 각각 한 개씩 정해져 있다.

정점은 총 N 개 있으며, 간선은 총 M 개 있다. N 개의 정점은 각각 1부터 N 까지의 번호가 부여되어 있다. 출발점은 S 번 정점이고, 도착점은 E 번 정점이다. 당신은 S 에서 출발해 E 에 도착하는 모든 경로들이 적어도 K 개의 지도를 거쳐가도록 지도를 설치해야 한다. S 에서 E 로 가는 경로가 하나도 존재하지 않는 경우에는, 지도를 하나도 설치하지 않아도 조건을 만족한 것으로 본다.

한 정점에는 지도를 한 개만 설치할 수 있으며, 각 정점마다 지도를 설치하는 데 드는 비용이 다르다. 정점 v 에 지도를 설치하는 데 필요한 비용은 양의 정수 C_v 로 주어진다. 출발점과 도착점에도 지도를 설치할 수 있다.

당신은 가능한 최소 비용으로 지도 설치를 완료하고 싶다. 조건을 만족하도록 지도를 설치할 수 있는지를 판별하고, 설치가 가능한 경우에는 최소 비용으로 지도를 설치하기 위해 어떤 정점들에 지도를 설치해야 하는지를 출력하라.

입력

첫 번째 줄에 정점 개수 N , 간선 개수 M , 최소 지도 개수 K 가 주어진다. ($2 \leq N \leq 200, 1 \leq M \leq 500, 1 \leq K \leq 5$)

그 다음 줄에는 출발점 S 와 도착점 E 가 주어진다. ($1 \leq S, E \leq N, S \neq E$)

그 다음 줄에는 i 번째 정점의 지도 설치 비용 C_i 를 나타내는 양의 정수 N 개가 정점 순서대로 공백으로 구분되어 주어진다. ($1 \leq C_i \leq 10\,000\,000, 1 \leq i \leq N$)

그 다음 M 개 줄에는 j 번째 간선이 연결하는 두 정점 u_j, v_j 가 주어진다. j 번째 간선은 u_j 번 정점에서 출발해서 v_j 번째 정점에 도달한다. ($1 \leq u_j, v_j \leq N, 1 \leq j \leq M$)

주어진 그래프에서 한 정점에서 출발해 동일한 정점에 도착하는 간선이 없음이 보장되며, 임의의 서로 다른 두 정점 쌍 (u, v) 에 대해, $u \rightarrow v$ 와 $v \rightarrow u$ 인 간선은 동시에 입력으로 주어질 수 있지만 $u \rightarrow v$ 는 최대 한 개, $v \rightarrow u$ 도 최대 한 개만 입력으로 주어질 수 있다.

주어진 그래프에서 S 에서 출발해 E 에 도착하는 경로가 존재하지 않을 수 있음에 유의하라.

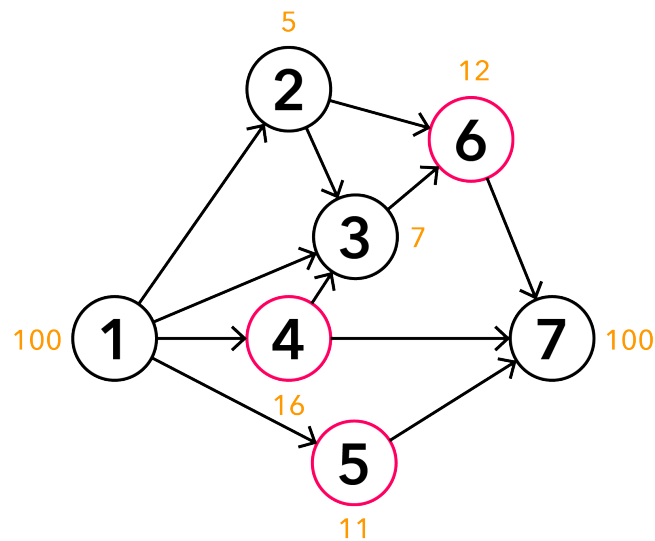
출력

조건을 만족하는 지도 설치가 불가능한 경우, 첫 번째 줄에 -1 을 출력한다.

조건을 만족하도록 지도 설치가 가능한 경우, 첫 번째 줄에 설치해야 하는 정점의 개수 P 를 출력한다. ($0 \leq P \leq N$) 두 번째 줄에는 조건을 만족시키기 위해 지도를 설치해야 하는 정점 P 개를 공백으로 구분하여 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
3 2 5 1 3 1 60 35 1 2 2 3	-1
7 11 1 1 7 100 5 7 16 11 12 100 1 2 1 3 1 4 1 5 2 3 2 6 3 6 4 3 4 7 5 7 6 7	3 5 4 6



문제 F. 애완 트리

시간 제한 8 초
메모리 제한 1024 MB

종영이는 애완 트리를 키우고 있다. 이 트리는 N 개의 정점이 있으며 i 번째 간선은 두 정점 u_i 와 v_i 를 잇는다.

요즘 트리에게도 사춘기가 와서, 번덕이 심해 간선들의 길이가 매일 바뀐다. i 번째 간선의 길이는 $[L_i, R_i]$ 범위의 자연수의 값들 중 하나를 가질 수 있다.

트리의 번덕에 짜증이 난 종영이는 참을성이 부족해 트리를 팔아버리기로 했다. 트리는 지름이 S 이상 E 이하이면 크기가 적절한 좋은 트리라 생각되어 비싸게 취급된다. 트리의 지름은 트리 상에서 임의의 두 정점 사이의 거리 중 최댓값을 뜻한다.

간선들의 가능한 길이들의 조합은 총 $\prod_{i=1}^{N-1} (R_i - L_i + 1)$ 가지임을 알 수 있다. 종영이가 트리를 비싸게 팔아치우는 것을 도와주기 위해 가능한 모든 조합에 대해 트리의 지름이 S 이상 E 이하인 경우의 수를 구하여라.

입력

첫 줄에 N, S, E 가 주어진다. ($2 \leq N \leq 400, 1 \leq S \leq E \leq 10^9$)

그 후 $N-1$ 개의 줄에 걸쳐 트리의 간선들의 정보가 주어진다. 정보는 u_i, v_i, L_i, R_i 순서로 주어진다.

($1 \leq u_i, v_i \leq N, 1 \leq L_i \leq R_i \leq 400$)

출력

트리의 지름이 S 이상 E 이하인 경우의 수를 $10^9 + 7$ 로 나눈 나머지를 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
4 3 14 1 2 1 10 1 3 1 10 1 4 1 10	573
8 17 31 3 4 5 5 1 8 8 8 8 2 8 10 6 7 8 10 6 3 9 10 8 6 7 10 7 5 1 10	159

이 페이지는 공백입니다

문제 G. 그건 망고가 아니라 고양이에요

시간 제한 3 초
메모리 제한 1024 MB

망고는 이하의 집에 사는 고양이이다. 이름의 유래는 집에 있던 망고주스와 색깔이 유사해서이다. 망고는 이 이름을 마음에 들어하는지 모르겠다.

삶은 힘들지만 고양이는 귀엽다. 그래서 몇몇 사람들은 망고를 칭송하기도 하며, ‘망고가 얼망고?’ 같은 말장난을 하거나, ‘망고 맛있겠다’ 같은 중의적인 농담을 펼치기도 한다. 이하는 망고가 망고라고 주장하지만 몇몇 사람들은 ‘그건 망고가 아니라 고양이에요’라 말하기에 이르렀고, 어쩌다보니 여기에서 이하는 끊임없는 PS 문제 창작 욕구에 의해 다음과 같이 문제를 만들게 되었다.

기본 문자열 M_0 와 규칙 문자열 S 가 있다고 하자. 이 때 양의 정수 i 에 대해 M_i 는 S 에 있는 모든 $\$$ 문자를 문자열 M_{i-1} 로 대체한 문자열로 정의된다. 문자열이 길지 않다면 몇 개는 손으로 만들어볼 수도 있다. M_0 를 ‘그건 망고가 아니라 고양이에요’라 하고, S 를 ‘그건 "\$"가 아니라 "\$"예요’라 하면 M_0, M_1, M_2 는 다음과 같다.

- M_0 : 그건 망고가 아니라 고양이에요
- M_1 : 그건 "그건 망고가 아니라 고양이에요"가 아니라 "그건 망고가 아니라 고양이에요"예요
- M_2 : 그건 "그건 "그건 망고가 아니라 고양이에요"가 아니라 "그건 망고가 아니라 고양이에요"예요"가 아니라 "그건 "그건 망고가 아니라 고양이에요"가 아니라 "그건 망고가 아니라 고양이에요"예요"예요

M_3, M_4 뿐만 아니라 $M_{1\,000}$ 도 똑같은 원리로 만들어낼 수 있다. 그러나 문자열의 길이가 너무 길어질 수 있기 때문에 일반적으로는 전체를 만들 수는 없다. 그러나 꼭 전체를 구할 필요는 없지 않은가? 이하는 이렇게 생성되는 문자열의 연속한 부분을 구해보고자 한다.

입력

첫 번째 줄에는 기본 문자열 M_0 가, 두 번째 줄에는 규칙 문자열 S 가 주어진다. 입력으로 들어오는 문자열은 다음 조건을 만족한다.

- 각 문자열의 길이는 1 이상 10^5 이하이다.
- 각 문자열의 모든 문자는 줄바꿈을 제외하고 ASCII code 값이 33 이상 126 이하이다. 즉, 제어 문자(control character)가 아닌 출력 가능한 문자(printable character)로만 구성되어 있다.
- M_0 에는 $\$$ (ASCII code 36) 문자가 없다.
- S 에는 $\$$ (ASCII code 36) 문자가 최소 하나 있다.

세 번째 줄에는 두 개의 양의 정수 k 와 Q 가 공백으로 구분되어 주어진다. ($1 \leq k \leq 10^5, 1 \leq Q \leq 10^5$)

네 번째 줄부터 Q 개의 줄에 걸쳐 질의가 주어진다. 이 Q 개의 줄 중 i 번째 줄에는 두 개의 정수 a_i 와 b_i 가 공백으로 구분되어 주어진다. ($1 \leq a_i \leq b_i \leq 10^{18}, b_i - a_i < 10^5$)

b_i 는 M_k 의 길이를 초과하지 않으며, $b_i - a_i + 1$ 의 합은 5×10^5 을 넘지 않는다.

출력

Q 개의 줄에 걸쳐 M_k 의 부분문자열을 출력한다.

이 중 i 번째 줄에는 M_k 의 a_i 번째 글자부터 b_i 번째 글자까지, 총 $b_i - a_i + 1$ 개의 문자를 출력한다.



입출력 예시

표준 입력(stdin)	표준 출력(stdout)
It's_a_cat,_not_a_mango It's_"\$",_not_"\$" 1 6 1 20 18 35 49 61 29 40 41 50 5 5	It's_"It's_a_cat,_no _not_a_mango",_not _not_a_mango" o",_not_"It' s_a_cat,_n -
Ad_finitum \$ 100000 4 1 10 1 2 4 10 5 8	Ad_finitum Ad finitum init

표준 입력(stdin)
THE_END \$_IS_NEVER_\$_IS_NEVER_\$ 88 5 1 7 3256 3257 67706 67710 111011 111017 999999999999999968 999999999999999993
표준 출력(stdout)
THE_END IS NEVER THE_END _THE_END_IS_NEVER_THE_END_

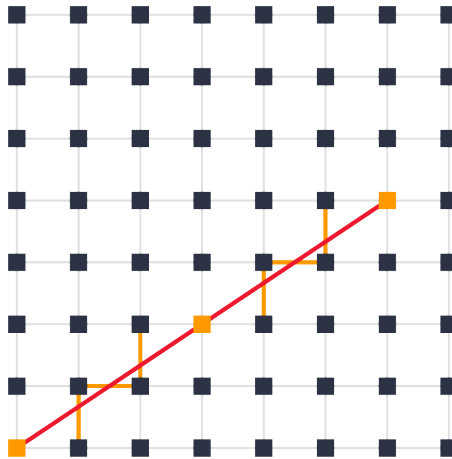
문제 H. 레이저 연구소

시간 제한 2 초
메모리 제한 1024 MB

아무도 모르는 외판 섬에서 ahgus89와 Sait2000은 레이저 연구를 하고 있다. 이 섬에는 다양한 연구 구역이 있고, 그 중 레이저 연구 구역은 $N \times M$ 모양의 직사각형 영역이다. 이 영역은 NM 개의 1×1 정사각형으로 쪼개져 있다. 각 정사각형의 꼭짓점에는 건물이 하나씩 세워져 있으며, 각 정사각형의 변에는 건물을 잇는 벽이 하나씩 세워져 있다. 따라서 총 $(N+1)(M+1)$ 개의 건물과 $2NM+N+M$ 개의 벽이 있다.

ahgus89와 Sait2000은 레이저의 성능을 시험하기 위해 건물에서 건물로 레이저를 쏘아 보려고 한다. 이들은 가능한 모든 시작점과 끝점에 대해 레이저를 쏠 것이다. 하지만 레이저는 직선으로만 나가기 때문에 건물과 벽이 레이저를 가로막을 수 있다.

ahgus89는 출력을 강하게 하여 건물과 벽에 구멍을 내어서라도 실험을 하려고 한다. 시작점과 끝점에 있는 건물을 포함하여, 경로 상에 있는 모든 건물과 벽에 구멍이 뚫린다. 레이저가 정확히 건물을 통과할 경우, 그 건물에 붙어 있는 벽에는 구멍이 뚫리지 않는다. 다만 레이저를 x 축이나 y 축에 평행하게 쏘 경우 아예 벽이 무너져 버리기 때문에 이런 경우는 제외하고 레이저를 쏠 것이다.



위 그림은 $N=7, M=7$ 인 경우 $(0,0)$ 에서 $(6,4)$ 로 레이저를 쏜 모습이다. 3개의 건물과 6개의 벽에 구멍이 뚫린다. 또한, 레이저를 한 번 쏘고 나면 구멍이 뚫린 건물과 벽을 전부 수리할 것이다. 건물을 수리하는 데에는 A 원, 벽을 수리하는 데에는 B 원이 든다. 수리를 완료해야 다음 레이저를 쏠 수 있다.

ahgus89의 계획을 들은 Sait2000은 수리비가 얼마나 들 지 생각하며 고민에 빠졌다. Sait2000을 위해 총 수리비를 구해주자!

입력

첫 번째 줄에 N, M, A, B 가 공백으로 구분되어 주어진다. ($1 \leq N, M, A, B \leq 10^9$)

출력

실험을 완료한 뒤의 총 수리비를 출력한다. 단, 답이 너무 커질 수 있으니 $10^9 + 7$ 로 나눈 나머지를 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
1 1 5 6	40
2 2 3 1	244
14 15 134 187	89892000

노트

예제 1의 상황은 다음과 같다.

(0,0)에서 (1,1)로 쏘고 나면 2개의 건물을 수리한다.

(1,0)에서 (0,1)로 쏘고 나면 2개의 건물을 수리한다.

(0,1)에서 (1,0)로 쏘고 나면 2개의 건물을 수리한다.

(1,1)에서 (0,0)로 쏘고 나면 2개의 건물을 수리한다.

따라서 총 8개의 건물을 수리하므로, 총 수리비는 40원이다.

예제 2에서 (0,0)에서 (2,2)로 쏘고 나면 3개의 건물만을 수리함에 유의하라.

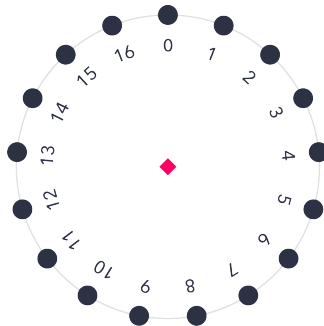
문제 I. 빛의 전사 크리스퍼

시간 제한 1 초
메모리 제한 1024 MB

여기는 평화로운 알고리즘 나라. 알고리즘 나라의 외곽에는 원 모양의 울타리가 쳐져 있다.

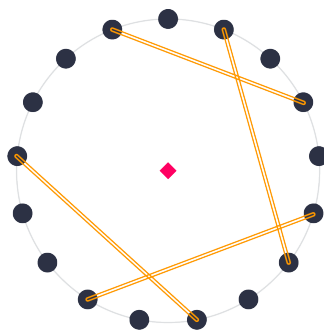
원 둘레에는 같은 간격으로 L 개의 기둥이 세워져 있고, 각 기둥은 0번부터 $L-1$ 번까지 시계방향 순으로 번호가 부여된 상태이다. 즉 0번 기둥의 시계방향에는 1번 기둥이 이웃해 있고, 반시계방향으로 $L-1$ 번 기둥이 이웃해 있다. (L 은 홀수)

편의상 0번 기둥이 12시 방향에 위치하도록 회전시킨 상태로 생각하자.



알고리즘 나라에는 나라의 수호자 크리스퍼가 살고 있다. 그 어떤 악인도 크리스퍼를 보게 되면 “그저 빛...”을 외치며 마음이 깨끗하게 정화된다. 때문에 이 마을에서는 그를 빛의 전사 **크리스퍼**라고 부르고, :fan: 클럽 등을 운영하며 그를 찬양하고 있다.

오늘은 알고리즘 나라에서 UCPC 2020 본선 대회가 열리는 날. 평화롭게 UCPC 대회를 진행하던 어느 순간, 사악한 악당 pichulia가 알고리즘 나라를 습격했다! 평소 알고리즘 나라에 악감정이 있던 pichulia는 기둥간에 N 개의 고무줄을 연결하는 장난을 쳤다. pichulia가 설치한 N 개의 고무줄 중 하나라도 남아 있으면 **출력 초과**를 받게 되므로, 이들을 모두 끊어내야 한다.

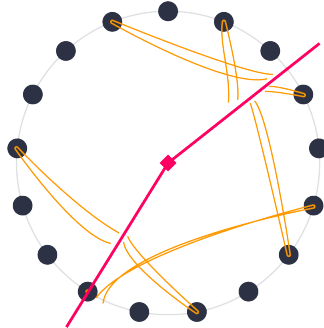


알고리즘 나라의 수호자 크리스퍼는 이러한 장난을 그냥 두고 볼 수 없었다. 따라서 크리스퍼의 필살기인 **루비빔**을 발사해서 모든 고무줄을 끊어내고자 한다.

크리스퍼의 위치는 원의 중심에 있으며, 원의 중심에서 이동하지 않고 고정된 상태로 루비빔을 발사한다. 루비빔은 원의 중심에서 시작하는 반직선 형태로 표현할 수 있다. 이 반직선과 교차하면 빔에 맞은 것으로 취급되고, 빔에 한 번이라도 맞은 고무줄은 끊어진다. 고무줄을 고정시킨 기둥에 빔을 맞은 경우도 고무줄이 끊어진다. 빔의 두께와 고무줄의 두께는 무시할 수 있을 정도로 얇다.

루비빔은 한 번 발사하는데 생각보다 많은 체력을 요구한다. 따라서 루비빔을 최소한으로 발사하는 것이 좋다.

고무줄을 모두 끊어내기 위해 필요한 루비빔 발사 횟수의 최소값을 구해서 **크리스퍼**에게 알려주자!



입력

첫 번째 줄에는 2개의 정수 N 과 L 이 공백을 사이에 두고 주어진다. N 은 고무줄의 개수이고 L 은 원에 있는 기둥의 개수이다. ($1 \leq N \leq 10^5$, $3 \leq L \leq 10^9$, L 은 홀수)

그 후 두 번째 줄부터 $N+1$ 번째 줄까지 N 줄에 걸쳐서 고무줄의 정보를 나타내는 2개의 정수 s, e 가 공백을 사이에 두고 주어진다. 이는 s 번 기둥과 e 번 기둥을 연결하는 고무줄을 설치했음을 의미한다. ($0 \leq s, e \leq L-1$, $s \neq e$) 두 개 이상의 고무줄이 서로 교차할 수 있다.

L 은 홀수이므로, 원의 중심을 지나는 고무줄은 존재하지 않는다.

출력

N 개의 고무줄을 모두 끊어내기 위해 필요한 루비빔의 최소 발사 횟수를 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
4 17 3 16 1 6 10 5 8 13	2
5 13 10 2 12 0 0 12 1 10 2 9	1
7 27 9 3 21 2 23 7 1 3 25 7 2 18 23 18	2
3 3 0 1 1 2 2 0	2

문제 J. 관광 사업

시간 제한 5 초
메모리 제한 1024 MB

옥토끼나라는 N 개의 도시를 잇는 $N-1$ 개의 도로로 이루어진 나라다. 어떤 도시에서도 원하는 다른 도시로 도로만을 통해 이동할 수 있다. 즉, 옥토끼나라는 트리 구조를 이룬다. 각 도로는 자연수 길이를 가지고 있으며, 두 도시 간의 거리는 두 도시를 잇는 단순 경로 위의 도로의 길이의 합이다.

옥토끼나라의 새로운 관광 사업으로 두 도시 X 와 Y 를 골라, 두 도시간에 관광 자매결연 관계를 맺을 것이다. 자매결연을 맺으면 두 도시의 사람들이 서로 관광을 하기 위해 이동할 것이다. 그렇기에 $D(X, Y)$ 를 X 와 Y 사이의 거리, C_X 와 C_Y 를 각각 X 와 Y 에 거주하는 인구 수라고 하면 교통료로 $(C_X + C_Y) \times D(X, Y)$ 의 수익을 얻을 수 있다.

옥토끼나라는 요즘 격변을 겪고 있어 도시의 인구 수가 계속 바뀌고, 자매결연 계획에 참가하려는 도시들도 상황에 따라 다양하기 때문에 다양한 상황에서 자매결연 관계를 맺을 도시들을 구해야 한다.

Q 개의 자매결연 계획이 주어진다. 각 계획은 X 의 후보 도시의 집합 A , Y 의 후보 도시의 집합 B 가 주어지며, 각 후보 도시의 인구 수 C_u 가 주어진다. A 와 B 에 동시에 포함되는 도시는 없다.

각 자매결연 계획마다, X 와 Y 를 정해서 얻을 수 있는 최대의 교통료 수익을 구해야 한다.

입력

첫 줄에 N 과 Q 가 공백으로 구분되어 주어진다. ($1 \leq N \leq 300\,000$, $1 \leq Q \leq 100\,000$)

그 다음 $N-1$ 개의 줄에 걸쳐 도로의 정보 u, v, d 가 공백으로 구분되어 주어진다. 이는 i 번째 도로가 도시 u 와 도시 v 를 이으며 길이는 d 라는 뜻이다. ($1 \leq u, v \leq N$, $1 \leq d \leq 30$)

이후 Q 개의 자매결연 계획이 다음과 같은 형식으로 주어진다.

- 첫 줄에 N_A 와 N_B 가 공백으로 구분되어 주어진다. N_A 는 A 의 크기, N_B 는 B 의 크기다. ($1 \leq N_A, N_B, N_A + N_B \leq N$)
- 이후 N_A 개의 줄에 걸쳐 u 와 p 가 공백으로 구분되어 주어진다. 이는 도시 u 가 집합 A 에 속하며 거주하는 인구 수가 p 명, 즉 $C_u = p$ 라는 뜻이다. ($1 \leq u \leq N$, $1 \leq p \leq 3 \times 10^6$)
- 이후 N_B 개의 줄에 걸쳐 v 와 q 가 공백으로 구분되어 주어진다. 이는 도시 v 가 집합 B 에 속하며 거주하는 인구 수가 q 명, 즉 $C_v = q$ 라는 뜻이다. ($1 \leq v \leq N$, $1 \leq q \leq 3 \times 10^6$)

$\sum_{i=1}^Q (N_A + N_B)$ 는 200 000 이하다.

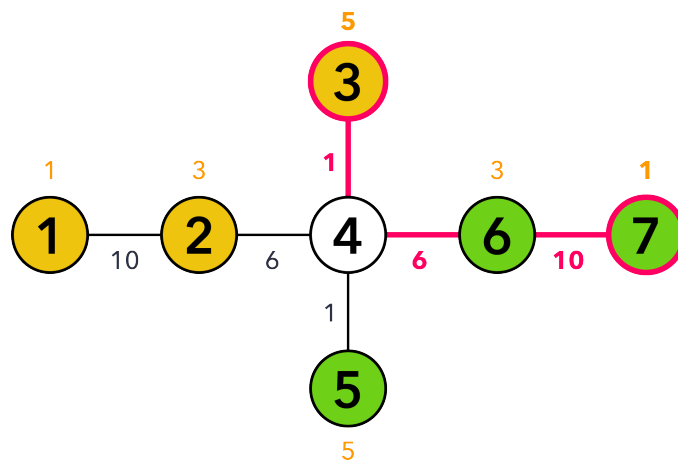
출력

한 줄에 하나씩 순서대로 각 자매결연 계획의 최대 교통료 수익을 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
<pre> 3 3 1 2 1 2 3 1 1 2 1 1 2 2 3 3 1 2 2 2 1 1 3 3 1 2 3 3 1 1 2 2 </pre>	<pre> 8 5 8 </pre>
<pre> 7 1 1 2 10 3 4 1 4 5 1 2 4 6 4 6 6 6 7 10 3 3 1 1 2 3 3 5 5 5 6 3 7 1 </pre>	<pre> 102 </pre>

노트



예제 2에서, $X = 3$ 이고 $Y = 7$ 인 경우 교통료 수익이 $(5 + 1) \times 17 = 102$ 로 최적이다. $X = 1$ 이고 $Y = 5$ 인 경우 또한 최적이다.

문제 K. 데이터 제작

시간 제한 1 초
메모리 제한 1024 MB

“정휘야, 데이터 만들어야지”

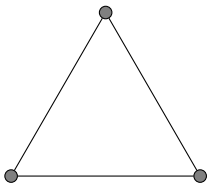
정휘는 원래 UCPC 2020 Call for Tasks에 점과 선분이 등장하는 재미있는 기하 문제를 제출하려고 했지만, 데이터를 너무 만들기 귀찮은 나머지 출제를 포기했다. UCPC 2020에 문제를 꼭 출제하고 싶어 고민하고 있던 정휘는 아주 좋은 아이디어를 냈다. 바로 대회 참가자들에게 데이터를 만들게 하는 것이다! 원래는 정휘가 해야 할 일이었지만, 이제는 여러분이 좌표 평면상에 점 N 개와 선분 M 개를 적절히 배치해서 K 개의 영역이 있는 데이터를 만들어야 한다.

영역은 평면 상의 빈 공간이며, 선분으로 모든 면이 둘러 쌓여 있어야 한다. 영역은 다른 영역으로 둘러 쌓여질 수도 있다. 좌표 범위가 너무 넓으면 계산하기 너무 힘들기 때문에, 79brue의 팬인 정휘는 모든 점의 좌표 범위를 79 이하의 자연수로 제한하기로 했다!

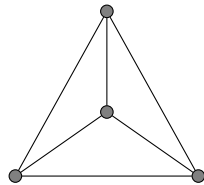
여러분은 아래 조건을 모두 만족시키는 데이터를 만들어야 한다.

- 각 점의 x, y 좌표는 1 이상 79 이하의 자연수가 되어야 한다.
- 모든 점의 위치는 서로 달라야 한다.
- 같은 두 점을 잇는 선분이 여러 개 존재하면 안 된다.
- 선분은 서로 다른 두 점을 이어야 한다.
- 서로 다른 두 선분은 끝점을 제외한 곳에서 교차하면 안 된다.
- 선분의 양 끝점을 제외한 점은 선분 위에 있으면 안 된다.

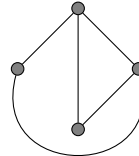
아래에서 (a)는 점 3개, 선분 3개로 1개의 영역을 만든 그림이고, (b)는 점 4개, 선분 6개로 3개의 영역을 만든 그림이다. (c)는 곡선이 존재하기 때문에, (d)는 교차하는 선분들이 존재하기 때문에 올바르지 않은 출력이다.



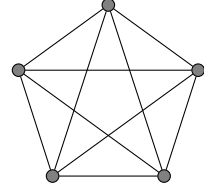
(a) 점 3개와 선분 3개로 영역 1개를 만든 그림



(b) 점 4개와 선분 6개로 영역 3개를 만든 그림



(c) 곡선이 존재하여 조건에 맞지 않는 그림



(d) 교차하는 선분이 존재해 조건에 맞지 않는 그림

입력

첫 번째 줄에 배치해야 할 점의 개수, 선분의 개수와 만들어야 하는 영역의 개수를 나타내는 세 자연수 N, M, K 가 주어진다. ($3 \leq N \leq 3\,000$, $0 \leq M$, $0 \leq K$)

N 개의 점과 M 개의 서로 교차하지 않는 선분으로 K 개의 다각형을 만들 수 있는 경우만 입력으로 주어진다.

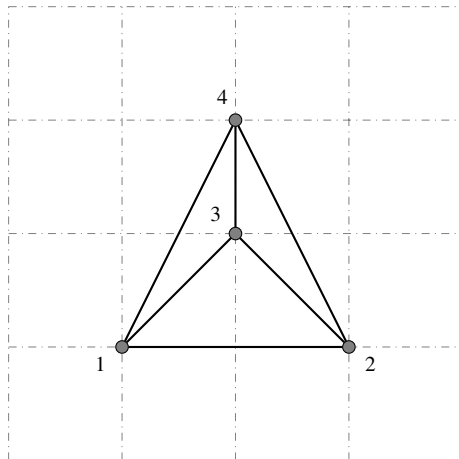
출력

첫째 줄부터 N 번째 줄까지 i 번째 줄에 i 번째 점의 좌표를 출력한다. $N+1$ 번째 줄부터 $N+M$ 번째 줄까지 각 선분이 몇 번째 점을 연결하는지 출력한다.

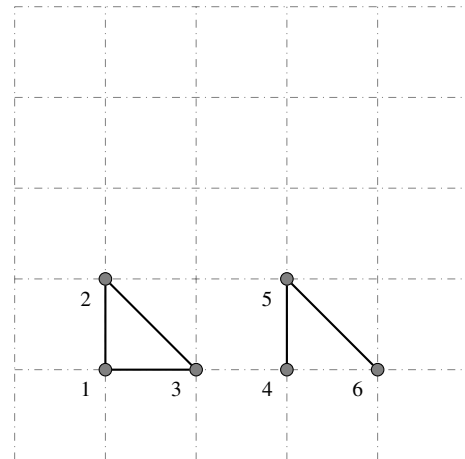
입출력 예시

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

노트



(a) 예제 1에 해당하는 그림



(b) 예제 2에 해당하는 그림

문제 L. 피자 배틀

시간 제한 5 초
메모리 제한 1024 MB

실버는 피자를 매우 빠르게 먹는다. 실버의 속도에 대적할 수 있는 유일한 상대는 그의 동생 브론즈이다. 두 사람의 피자 먹는 속도는 초당 1 단위로 동일하다.

실버와 브론즈는 서로의 능력을 과시하기 위해 게임을 하기로 했다. 게임은 N 개의 조각으로 이루어진 원형의 피자에서 진행되며, 각각의 조각의 크기는 1 이상 100 이하의 정수다. 두 사람의 목표는 피자를 최대한 많이 먹는 것이다.

실버부터 시작하여 피자 조각을 고르는데, 첫 번째 조각은 아무렇게나 골라도 되지만, 그 다음 조각부터는 바깥쪽에 있는 조각만을 골라야 한다. 즉, 고르려고 하는 조각의 양 옆 중 한 곳 이상이 비어 있어야 한다.

처음에 실버가 피자 조각을 고른 뒤 0.5초 후에 브론즈가 피자 조각을 고른다. 또한, 두 플레이어는 선택한 피자 조각을 다 먹은 직후에 다음 피자 조각을 골라야 한다. 이 때 각자가 피자 조각을 고르는 시간은 무시한다.

피자의 크기가 순서대로 1, 2, 3, 4일 때 가능한 게임 양상의 예시는 다음과 같다.

$t = 0$: 실버가 3을 고르고 먹기 시작함.

$t = 0.5$: 브론즈는 2와 4 중 하나를 골라서 먹을 수 있음. 2를 고르고 먹기 시작함.

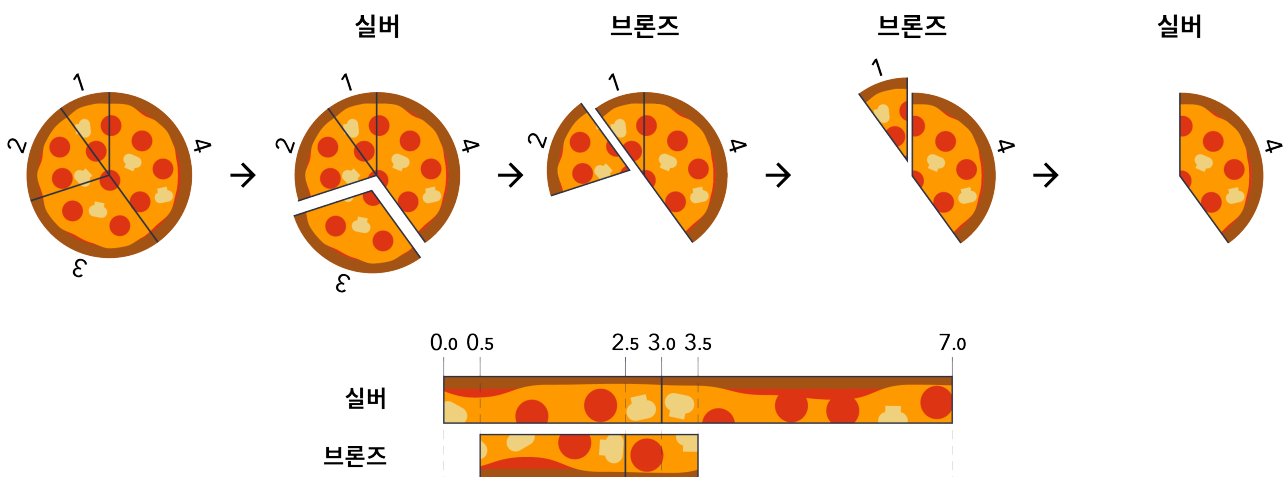
$t = 2.5$: 브론즈가 2짜리 조각을 다 먹음. 1과 4 중 1을 골라서 먹기 시작.

$t = 3$: 실버가 3짜리 조각을 다 먹음. 4를 골라서 먹기 시작.

$t = 3.5$: 브론즈가 1짜리 조각을 다 먹었지만, 고를 피자 조각이 없으므로 더 이상 피자를 못 먹음.

$t = 7$: 실버가 4짜리 조각을 다 먹음.

최종 결과: 실버 7, 브론즈 3



예시에 있는 두 사람의 플레이는 최선의 플레이가 아닐 수 있음에 유의한다. 피자가 주어지면, 두 사람이 최선의 전략으로 플레이했을 때 실버가 먹게 되는 피자의 양을 구하는 프로그램을 작성하여라.

입력

첫 번째 줄에 피자 조각의 수 N 이 주어진다. ($1 \leq N \leq 1000$)

두 번째 줄에 N 개의 피자 조각의 크기가 시계 방향 순서대로 주어진다. 피자 조각의 크기는 1 이상 100 이하의 정수다. 원형이므로 1번 피자 조각과 N 번 피자 조각은 연결되어 있다.

출력

두 사람이 최선의 전략으로 플레이했을 때 실버가 먹게 되는 피자의 양을 출력한다.

입출력 예시

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