

문제 배치는 다음과 같습니다.

번호	문제명	초	중	고
A	와글와글 송고한	O		
B	보너스 점수	O		
C	이건 꼭 풀어야 해!	O	O	
D	무한부스터	O	O	
E	우울한 방학	O		
F	다이나믹 롤러	O		
G	핑거 스냅	O		
H	이진수 변환		O	O
I	백도어		O	O
J	FLEX		O	
K	통신망 분할		O	
L	깃발춤			O
M	일하는 세포			O
N	트리의 외심		O	O
O	시간 끌기			O
P	가장 높고 넓은 성			O

문제 A. 와글와글 송고한

시간 제한: 1 초

메모리 제한: 512 MB

송고한 알고리즘 캠프가 다가오고 있고 방학이 되어서까지도 각 대학들의 협업은 계속되고 있다. 그럼에도 불구하고 운영진들과 강사진들이 각자의 일정 때문에 바빠 계획에 차질이 조금씩 생기고 있다. 송고한 알고리즘 캠프의 대표인 창호는 효율적인 일처리를 위해 엄정한 평가를 내리기로 하였다.

창호는 송고한 알고리즘 캠프의 구성원인 송실대학교(Soongsil University), 고려대학교(Korea University), 한양대학교(Hanyang University)의 참여도를 수치화하였다. 창호가 보기에 세 대학교의 참여도의 합이 100 이상이면 일처리가 잘 되고 있기에 안심할 수 있지만, 100 미만이면 창호는 참여도가 가장 낮은 대학의 동아리에게 무언의 압박을 넣을 예정이다. 송고한 알고리즘 캠프의 성공을 빌며 창호의 고민을 해결해주자.

입력 형식

입력의 첫 줄에는 송실대학교의 참여도, 고려대학교의 참여도, 한양대학교의 참여도를 의미하는 세 자연수 S , K , H 가 공백으로 구분되어 주어진다. ($0 \leq S, K, H \leq 100$)

세 대학의 참여도는 모두 다르다.

출력 형식

첫 번째 줄에 일처리가 잘 되고 있어 무언의 압박이 필요가 없으면 (따옴표를 제외하고) "OK"를 출력한다.

그 외에는 첫 번째 줄에 무언의 압박이 필요한 동아리가 속한 대학의 영문 이름의 첫 단어를 출력한다. 영문 이름 표기는 지문에 나온 것을 따른다.

예제

표준 입력	표준 출력
31 41 59	OK
1 2 3	Soongsil
19 8 9	Korea
45 33 21	Hanyang

문제 B. 보너스 점수

시간 제한: 1 초

메모리 제한: 512 MB

송고한 알고리즘 캠프 퀴즈 타임이 시작되었다! PS 기초, 동적 계획법, 파라메트릭 서치, 욕제의 생일, 탐색, 그리디, 최단경로 알고리즘, 구테기컵, 서로소 집합, 최소 신장 트리, 최소 공통 조상, 세그먼트 트리, 코드포스에서 C++로 높은 수준의 난수를 생성하는 방법, 최대 유량, 볼록 껍질, 스타트링크 사무실에 있는 게임용 컴퓨터의 RAM의 총 용량 등등 수많은 주제를 총망라하고 있는 이 미니 대회는 수많은 참가자들의 도전으로 오늘도 빛나고 있고, 제출된 OX표의 개수는 셀 수 없을 정도이다.

운영진들은 이 OX표들을 채점하고, 점수를 계산해낸 다음, 시상식을 진행하며 화기애애하게 행사를 마무리해야 한다. 송고한 알고리즘 캠프 퀴즈 타임에서 OX표의 점수는 다음과 같이 계산된다.

- OX표에 N 개의 문제들이 있을 때, 1번 문제, 2번 문제, ..., N 번 문제 순으로 채점된다.
- 문제는 뒤로 갈수록 어려워지기 때문에, i 번 문제의 기본 점수는 i 점이다.
 - 문제를 맞히면 그 문제의 기본 점수(즉 i 번 문제의 경우 i 점)를 획득하며, 틀리면 얻지 못한다.
- 기본 점수와 별개로, '보너스 점수'라는 값이 존재한다. 이는 처음에는 0점이다.
 - 문제를 맞히면 그 때의 '보너스 점수'를 획득하고, '보너스 점수'의 값이 1점 증가한다.
 - 문제를 틀리면 '보너스 점수'를 얻지 못하고, '보너스 점수'의 값이 0점으로 초기화된다.

민성이는 얼떨결에 송고한 알고리즘 캠프 퀴즈 타임의 OX표를 채점해야 하는 업무를 맡게 되었다. 수많은 OX표를 볼 생각에 머리가 지끈거리는 민성이는 프로그램을 세워 이를 자동화하려고 한다. 시상식까지 4시간밖에 남지 않은 민성이를 도와 점수를 계산해주자.

입력 형식

첫 번째 줄에 OX표의 길이인 자연수 N 이 주어진다. ($1 \leq N \leq 10,000$)

두 번째 줄에 OX표를 의미하는 문자열 S 가 주어진다. S 는 0(알파벳 대문자 O, ASCII 코드 79)와 x(알파벳 대문자 X, ASCII 코드 88)로만 구성되어 있으며, 길이는 N 이다.

문자열 S 의 i 번째 글자가 0이면 해당 참가자가 i 번째 문제를 맞혔음을 의미하고, x이면 틀렸음을 의미한다.

출력 형식

첫 번째 줄에 입력으로 들어온 OX표의 점수를 출력한다.

예제

표준 입력	표준 출력
8 X00X00X	26

설명

민성이는 다음과 같이 총 26점을 받게 된다.

- 1번 문제를 틀렸으므로 점수를 얻지 못하고 보너스 점수가 0점으로 초기화된다. (총 점수 0점)
- 2번 문제를 맞혔으므로 기본 점수 2점과 보너스 점수 0점을 획득하며, 보너스 점수가 1점으로 증가한다. (총 점수 2점)
- 3번 문제를 맞혔으므로 기본 점수 3점과 보너스 점수 1점을 획득하며, 보너스 점수가 2점으로 증가한다. (총 점수 6점)
- 4번 문제를 맞혔으므로 기본 점수 4점과 보너스 점수 2점을 획득하며, 보너스 점수가 3점으로 증가한다. (총 점수 12점)
- 5번 문제를 틀렸으므로 점수를 얻지 못하고 보너스 점수가 0점으로 초기화된다. (총 점수 12점)
- 6번 문제를 맞혔으므로 기본 점수 6점과 보너스 점수 0점을 획득하며, 보너스 점수가 1점으로 증가한다. (총 점수 18점)
- 7번 문제를 맞혔으므로 기본 점수 7점과 보너스 점수 1점을 획득하며, 보너스 점수가 2점으로 증가한다. (총 점수 26점)
- 8번 문제를 틀렸으므로 점수를 얻지 못하고 보너스 점수가 0점으로 초기화된다. (총 점수 26점)

문제 C. 이걸 꼭 풀어야 해!

시간 제한: 1 초

메모리 제한: 512 MB

송실꿀 높은 언덕 깊은 골짜기에 출제로 고통 받는 옥제가 살고 있다!

옥제는 또 출제를 해야 해서 단단히 화가 났다. 그래서 옥제는 길이 N 짜리 수열 A 를 만들고, A 를 비내림차순으로 정렬해서 수열 B 를 만들어 버렸다!! 여기서 B 를 출력하기만 하면 문제가 너무 쉬우니까 하나만 더 하자. 아래와 같은 질문이 무려 Q 개나 주어진다!! (ㅎㅎ;; 스스.. ㅋㅋ!!)

- **L R**: $B_L + B_{L+1} + \dots + B_{R-1} + B_R$ 을 출력한다.



Figure 1: 모든 참가자가 문제를 풀 수 있을 것이라고 기대하는 옥제의 표정

옥제의 질문에 답하고 함께 엠티를 떠나자!!

입력 형식

첫 번째 줄에 수열 A 의 길이 N 과 질문의 개수 Q 가 공백으로 구분되어 주어진다. ($1 \leq N, Q \leq 300,000$)

두 번째 줄에 N 개의 정수 A_1, A_2, \dots, A_N 이 공백으로 구분되어 주어진다. A_i 는 수열 A 의 i 번째 수이다. ($1 \leq A_i \leq 1,000$)

세 번째 줄부터 Q 개의 줄에 걸쳐 옥제의 질문을 의미하는 두 수 L, R 이 공백으로 구분되어 주어진다. ($1 \leq L \leq R \leq N$)

주어지는 모든 입력은 자연수이다.

출력 형식

Q 개의 줄에 걸쳐, 질문의 답을 순서대로 각각 출력한다.

예제

표준 입력	표준 출력
5 6 2 5 1 4 3 1 5 2 4 3 3 1 3 2 5 4 5	15 9 3 6 14 9
5 3 2 5 1 2 3 1 3 2 3 1 5	5 4 13

설명

첫 번째 예제에서 [2, 5, 1, 4, 3]을 비내림차순으로 정렬하면 [1, 2, 3, 4, 5]이다.

두 번째 예제에서 [2, 5, 1, 2, 3]을 비내림차순으로 정렬하면 [1, 2, 2, 3, 5]이다.

참고 사항

비내림차순은 원소가 감소하지 않는 (같거나 증가하는) 순서를 말한다.

```
while (Q--) {
    int sum = 0, L, R;
    scanf("%d %d", &L, &R);
    for (int i = L; i <= R; i++) {
        sum += a[i];
    }
    printf("%d\n", sum);
}
```

위와 같이 각 질문마다 반복문을 매번 돌려서 답을 계산하면, 시간복잡도가 $O(QN)$ 이 되므로 시간 초과를 받게 된다. 다른 방법을 이용해 문제를 해결해야 한다.

문제 D. 무한부스터

시간 제한: 1 초

메모리 제한: 512 MB

카트라이더를 처음 시작하는 카린이 정범이는 어려운 조작법에 실망감이 커져가고 있다. 드리프트, 순간 부스터, 커팅, 톱톡이 등등 어려운 테크닉에 질린 정범이는 그나마 쉬운 ‘송고한 무한부스터 모드’에 도전해보려고 한다.

‘송고한 무한부스터 모드’는 크기 $N \times M$ 의 직사각형 모양의 맵에서 진행되며, 맵 전체가 단위 격자로 구성되어 있다. 기존의 ‘무한부스터 모드’와는 다르게, 모든 격자 안에는 특정 개수의 부스터 아이템이 위치한다. 이 모드에서 플레이의 방식은 다음과 같다.

처음에 플레이어의 카트바디는 출발지점인 1행 1열에 위치하며, 멈춰 있는 상태이고, 보유하고 있는 부스터 아이템의 개수는 0개이다. 목표는 도착지점인 N 행 M 열의 격자에 도달하는 것이며, 도달하는 즉시 게임이 종료된다. 카트바디가 격자에 멈추어 있을 때, 격자에 놓여있는 부스터 아이템을 자동으로 전부 습득하게 된다. 이 과정에서 x 개를 습득했다면 한 방향을 정해 오른쪽으로 최대 x 칸을 가거나, 아래쪽으로 최대 x 칸을 이동할 수 있으며, 1칸 단위로 이동하게 된다. 예를 들어 부스터 아이템을 3개 습득했을 때, 오른쪽으로 2칸 이동이나 아래쪽으로 3칸 이동은 가능하지만, 오른쪽으로 1칸 이동 후 아래로 2칸 이동이나 왼쪽으로 1칸 이동이나 아래쪽으로 2.718칸 이동은 불가능하다. **이동 후 멈추면서 보유하고 있던 부스터 아이템은 모두 소진된다.**

이동중에 멈추지 않고 지나치는 격자의 부스터 아이템은 습득할 수 없으며, 카트바디는 맵을 벗어나는 방향으로 움직일 수 없다.

정범이는 ‘송고한 무한부스터 모드’에서 출발지점부터 도착지점까지 주행하면서 부스터 아이템을 획득하게 되는 격자의 개수를 최소화하고 싶다. 카린이 정범이를 도와주도록 하자.

입력 형식

첫 번째 줄에 맵의 세로 길이와 가로 길이를 나타내는 양의 정수 N 과 M 이 공백으로 구분되어 주어진다. ($1 \leq N, M \leq 300$)

두 번째 줄부터 N 개의 줄에 걸쳐 각 격자에 있는 부스터 아이템 개수인 M 개의 양의 정수 a_{ij} 가 공백으로 구분되어 주어진다. ($1 \leq a_{ij} \leq \max(N, M)$) a_{ij} 는 i 행 j 열의 격자에 있는 부스터 아이템 개수이다.

출발지점과 도착지점은 다르다.

출력 형식

첫 번째 줄에 정범이가 맵의 출발지점부터 도착지점까지 이동하면서 부스터 아이템을 획득하게 되는 격자의 최소 개수를 출력한다.

예제

표준 입력	표준 출력
4 5 1 1 4 1 3 3 4 1 3 2 1 1 5 3 2 5 3 1 1 1	3
5 4 2 4 2 3 1 1 1 3 2 1 2 2 1 4 4 1 1 2 2 1	3

설명

첫 번째 예제에서 3번 만에 이동하는 최적 경로는 다음과 같다.

- 1행 1열에서 1개의 부스터 아이템을 획득하고, 아래쪽으로 1칸 이동하여 2행 1열에서 멈춘다.
- 2행 1열에서 3개의 부스터 아이템을 획득하고, 아래쪽으로 2칸 이동하여 4행 1열에서 멈춘다.
- 4행 1열에서 5개의 부스터 아이템을 획득하고, 오른쪽으로 4칸 이동하여 4행 5열에서 멈춘다. 도착지점에 도달했기 때문에 게임이 종료된다.

두 번째 예제에서 3번 만에 이동하는 최적 경로는 다음과 같다.

- 1행 1열에서 2개의 부스터 아이템을 획득하고, 오른쪽으로 1칸 이동하여 1행 2열에서 멈춘다.
- 1행 2열에서 4개의 부스터 아이템을 획득하고, 아래쪽으로 4칸 이동하여 5행 2열에서 멈춘다.
- 5행 2열에서 2개의 부스터 아이템을 획득하고, 오른쪽으로 2칸 이동하여 5행 4열에서 멈춘다. 도착지점에 도달했기 때문에 게임이 종료된다.

문제 E. 우울한 방학

시간 제한: 1 초

메모리 제한: 512 MB

방학동안 기숙사에 홀로 남겨진 인호는 우울하고 고독하다. 다행히 인호는 M 일의 방학 동안 N 개의 약속이 잡혀있기에, 약속 날짜의 효율적인 배치를 통해 방학 내에 느낄 우울함의 합을 최소화하려고 한다.

인호의 기분은 정수로 표현 가능하며, 기분이 0 미만인 날에 (기분)² 만큼 우울함을 느낀다. 인호의 기분은 오늘 약속이 있다면 약속의 기대행복 값인 H_i 이며, 약속이 없으면 어제의 기분에서 1을 뺀 값이다.

인호는 하루에 최대 한 개의 약속을 소화할 수 있으며, N 개의 약속들의 순서는 주어진 순서대로여야 한다.

방학은 내일부터 시작이며, 오늘 인호의 기분은 0일 때, 약속을 적절히 배치하여 인호가 방학 동안 느낄 우울함의 합을 최소화하자.

입력 형식

첫 번째 줄에는 인호의 약속 개수인 자연수 N 과 방학의 일수인 자연수 M 이 공백으로 구분되어 주어진다. ($0 \leq N < M < 1000$)

두 번째 줄에는 N 개의 정수 H_1, H_2, \dots, H_N 이 공백으로 구분되어 주어진다. H_i 는 i 번째 약속의 기대행복 값이다. ($1 \leq H_i < 100$)

출력 형식

첫 번째 줄에 인호가 방학 동안 느낄 우울함의 합의 최솟값을 출력한다.

예제

표준 입력	표준 출력
3 10 2 2 1	2

설명

1일, 5일, 8일에 약속을 순서대로 배치하면, 4일과 10일에 각각 1만큼의 우울함을 느끼게 되어, 총 2만큼의 우울함을 느끼게 된다. 이보다 덜 우울함을 느끼게 만드는 방법은 존재하지 않는다.

문제 F. 다이내믹 롤러

시간 제한: 2 초

메모리 제한: 512 MB

페인팅 전문 회사 부키치 암즈는 거대한 페인팅용 롤러 “다이내믹 롤러”를 출시하였다. 이 신제품은 평범한 페인팅 롤러와 마찬가지로 굴려서 칠할 수 있지만, 손잡이를 세로로 휘둘러 잉크를 한번에 흘뿌릴 수도 있도록 설계되었다. 이러한 새로운 사용방법은 거대한 몸집과 맞물려 매우 역동적으로 보였기 때문에, 이 롤러의 이름은 다이내믹 롤러가 되었다. 평소 페인팅에 관심이 많던 멍미는 다이내믹 롤러의 매력에 흠뻑 빠져, 단숨에 다이내믹 롤러를 구매했다. 지금 당장 롤러를 시험해 보고 싶었던 멍미는 통로 일부분을 칠해보기로 했다.

통로는 $1 \times N$ 길이의 일자 형태를 가지고 있고, 통로의 바닥은 1×1 타일로 가득 차있다. 각 타일은 잉크지수 A_i 와 점도지수 B_i 를 가지고 있다. 타일이 제각각 다른 특성을 가지고 있기 때문에, 멍미는 세심하게 롤러를 휘둘러야만 한다. 멍미가 i 번째 타일 위에 서 있을 때, 멍미는 다이내믹 롤러로 현재 위치보다 오른쪽에 있으면서 점도지수가 서 있는 칸의 잉크지수 A_i 이하인 칸을 칠할 수 있다.

통로는 기본적인 관리가 되고 있기 때문에, 각 칸의 잉크지수 A_i 는 점도지수 B_i 이상이다. 그러나 깊은 통로는 관리에 어려움이 있기 때문에, 점도지수 B_i 는 항상 오름차순이다. 이런 상황 속에서 멍미가 통로의 각 타일에서 서 있을 때 다이내믹 롤러로 칠할 수 있는 최대의 칸 수를 구해보자.

입력 형식

첫 번째 줄에 통로의 길이인 자연수 N 이 주어진다. ($1 \leq N \leq 5 \times 10^5$)

두 번째 줄에 N 개의 정수 A_1, A_2, \dots, A_N 이 공백으로 구분되어 주어진다. A_i 는 i 번째 칸의 잉크지수를 의미한다. ($1 \leq A_i \leq 10^{18}$)

세 번째 줄에 N 개의 정수 B_1, B_2, \dots, B_N 이 공백으로 구분되어 주어진다. B_i 는 i 번째 칸의 점도지수를 의미한다. ($1 \leq B_i \leq 10^{18}, A_i \geq B_i$)

B_1, B_2, \dots, B_N 은 오름차순이다. 즉, $1 \leq i < j \leq N$ 을 만족하는 모든 정수 순서쌍 (i, j) 에 대해 $B_i \leq B_j$ 가 성립한다.

출력 형식

첫 번째 줄에 N 개의 정수 x_1, x_2, \dots, x_N 을 공백으로 구분하여 출력한다. x_i 는 i 번째 칸에 서서 다이내믹 롤러를 사용할 때 최대로 칠할 수 있는 칸의 개수이다.

예제

표준 입력	표준 출력
5 10 10 10 10 10 10 10 10 10 10	4 3 2 1 0
6 60 90 100 88 90 99 60 70 80 85 90 90	0 4 3 0 1 0

설명

첫 번째 예제 설명은 다음과 같다.

- 첫 번째 칸에서는 시작 지점의 잉크지수가 10이다. 두 번째 칸부터 다섯 번째 칸의 점도지수가 10이기에 다섯 번째 칸까지 칠할 수 있다. 그렇기에 총 4칸을 칠할 수 있다.
- 두 번째 칸도 똑같이 3칸, 세 번째 칸은 2칸, 네 번째 칸은 1칸, 다섯 번째 칸은 0칸 칠할 수 있다.

두 번째 예제 설명은 다음과 같다.

- 첫 번째 칸의 잉크지수는 60이다. 60보다 작거나 같은 점도지수를 가진 가장 오른쪽 칸이 첫 번째 칸이므로 답은 0칸이다.
- 두 번째 칸의 잉크지수는 90이다. 여섯 번째이자 마지막 칸의 점도지수가 90이므로 여섯 번째 칸까지 칠할 수 있으므로 답은 4칸이다.
- 세 번째 칸의 잉크지수는 100이다. 마찬가지로 여섯 번째 칸까지 칠할 수 있으므로 답은 3칸이다.
- 네 번째 칸의 잉크지수는 88이다. 88보다 작거나 같은 점도지수를 가진 가장 오른쪽 칸은 네 번째 칸이므로 답은 0칸이다.
- 다섯 번째 칸의 잉크지수는 90이다. 90보다 작거나 같은 점도지수를 가진 칸은 가장 오른쪽 칸은 여섯 번째 칸이므로 답은 1칸이다.
- 여섯 번째 칸의 잉크지수는 99이며, 99보다 작거나 같은 점도지수를 가진 칸은 가장 오른쪽 칸은 여섯 번째 칸이므로 답은 0칸이다.

문제 G. 핑거 스냅

시간 제한: 2 초

메모리 제한: 512 MB

[어벤져스] 시리즈를 보지 않은 사람이라도 ‘인피니티 건틀렛’이 무엇인지는 다들 알 것이다. 그래도 모르는 사람들을 위해 설명을 하자면, 인피니티 스톤이 모두 모인 인피니티 건틀렛을 끼고 손가락을 튕기면, 사용자가 원하는 것을 할 수 있다. 그러나 반동이 매우 심하기 때문에 그리 많이 사용할 수 없다.

정신 나간 수학자 Sonaht는 우연히 이 인피니티 건틀렛을 손에 넣게 된다. 그러나 이 인피니티 건틀렛에는 약간의 하자가 있어서, 핑거 스냅으로 할 수 있는 일이 몇가지 없다. 다음은, 핑거 스냅으로 할 수 있는 일을 나열한 것이다.

1. 전 우주의 생명체 수를 현재의 절반으로 한다.
2. 전 우주의 생명체 수를 현재의 $1/3$ 로 한다.
3. 전 우주의 생명체 수를 현재보다 하나 늘린다.
4. 전 우주의 생명체 수를 현재보다 하나 줄인다. 이미 전 우주의 생명체 수가 0이라면 할 수 없다.

첫 두 경우에서, 나누어 떨어지지 않으면 몫만 남기고, 나머지는 버린다.

Sonaht는 전 우주의 생명체 수를 목표치 A 이상 B 이하로 만들려고 한다. 그러나 역시나 정신 나간 수학자답게, A 이상 B 이하인 수 중 소수로 만들려 한다 (어쩌면 A 와 B 사이에 소수가 없을지도 모르지만 말이다.) 소수란, 서로 다른 약수가 1과 자기 자신밖에 없는 수를 의미한다. 그러나 인피니티 건틀렛은 반동이 심하기에, Sonaht는 최대한 적은 수의 핑거 스냅으로 이 목표를 달성하고자 한다. Sonaht가 최소 몇 번의 핑거 스냅을 해야 할지 구해보자.

입력 형식

첫 번째 줄에 테스트 케이스의 개수 T 가 주어진다. ($1 \leq T \leq 10$)

두 번째 줄부터 T 개의 줄에 걸쳐, 현재 전 우주의 생명체 수인 자연수 N 과, Sonaht의 목표 범위인 자연수 A , B 가 공백으로 구분되어 주어진다. ($2 \leq N \leq 1,000,000$, $2 \leq A \leq B \leq 100,000$)

출력 형식

매 줄마다, 각 테스트 케이스에서 Sonaht가 전 우주의 생명체 수를 목표범위 내의 소수로 만드는 데 필요한 최소한의 핑거 스냅의 횟수를 출력한다.

만약 목표범위 내의 소수로 만들 수 없다면, -1 을 출력한다.

매 테스트 케이스는 독립적으로 고려되어야 한다.

예제

표준 입력	표준 출력
5	1
9 2 4	15
100000 605 610	-1
300 14 16	0
7 7 10	12
98765 500 550	

설명

첫 번째 테스트 케이스에서는, 2번 동작 (전 우주의 생명체를 1/3로 줄임)을 하면 전 우주의 생명체의 수가 3이 되어, 2 이상 4 이하 소수가 된다. 그러므로 필요 횟수는 1이다.

세 번째 테스트 케이스에서는, 14 이상 16 이하 소수가 존재하지 않으므로, 아무리 핑거 스냅을 해도 목표를 달성할 수 없다. 따라서 -1을 출력한다.

네 번째 테스트 케이스에서는 핑거 스냅을 하지 않고도 목표가 달성되므로 필요 횟수는 0이다.

문제 H. 이진수 변환

시간 제한: 1 초

메모리 제한: 512 MB

당신에게 자연수 x_0 와 N 이 주어졌다. 지금부터 당신은 이 자연수 x_0 를 N 번의 ‘변환’을 통해 0으로 바꿀 것이다. 변환이란, 양의 정수를 이진법으로 표기했을 때, 1개 이상의 1을 0으로 바꾸는 작업이다. 예를 들어 9를 이진법으로 나타내면 $1001_{(2)}$ 인데, 9는 $0(0000_{(2)})$, $1(0001_{(2)})$, 또는 $8(1000_{(2)})$ 로 변환될 수 있다 (바뀐 자릿수는 밑줄로 표기되었다). 여러분의 목표는 x_i 를 변환하여 x_{i+1} 를 만드는 과정을 반복해, x_N 을 0으로 만드는 것이다.

위 조건을 만족하는 수열 $X = [x_0, x_1, x_2, \dots, x_N]$ 는 존재하지 않을 수도 있지만, 여러 개가 존재할 수도 있다. 만약 존재한다면, 각 수열별로 인접한 원소들의 차들의 집합 $D(X) = \{x_0 - x_1, x_1 - x_2, \dots, x_{N-1} - x_N\}$ 를 정의하자. 이 집합의 원소들의 최대값과 최소값의 차이를 최소화하도록, 수열 X 를 만들려고 한다. 즉, 가능한 모든 수열 X_i 중 ($D(X_i)$ 에 속한 원소의 최댓값 - $D(X_i)$ 에 속한 원소의 최솟값)이 최소가 되는 X_i 를 찾고자 한다.

이상해보일 수 있는 문제지만, 당신은 대답해야 한다. 과연 1초 안에 답할 수 있을까?

입력 형식

첫 번째 줄에 변환할 자연수와 변환 횟수를 의미하는 두 자연수 x_0 과 N 이 공백으로 구분되어 주어진다. ($1 \leq x_0 \leq 10^{16}, 1 \leq N \leq 50$)

출력 형식

만약 조건을 만족하는 수열이 존재하지 않으면 첫 번째 줄에 -1을 출력한다.

조건을 만족하는 수열이 존재한다면, 수열의 원소를 의미하는 N 개의 정수 x_1, x_2, \dots, x_N 을 공백으로 구분하여 출력한다.

조건을 만족하는 수열이 여러 개 존재한다면, 아무 것이나 출력해도 좋다.

예제

표준 입력	표준 출력
23 2	16 0
48 5	-1

설명

첫 번째 예제에서 $x_0 = 23 = 10111_{(2)}$, $x_1 = 16 = 10000_{(2)}$, $x_2 = 0 = 00000_{(2)}$ 로 수열을 구성하면 인접한 원소의 차의 최대값과 최소값의 차이는 $16 - 7 = 9$ 가 되며, 이보다 작게 만들 수 있는 방법은 존재하지 않는다. 또 다른 답으로는 $x_0 = 23, x_1 = 7, x_2 = 0$ 이 있다.

두 번째 예제에서 주어진 조건을 만족하는 x_1, x_2, x_3, x_4, x_5 는 존재하지 않는다.

문제 I. 백도어

시간 제한: 2 초

메모리 제한: 512 MB

유섭이는 무척이나 게으르다. 오늘도 할 일을 모두 미뤄둔 채 열심히 롤을 하던 유섭이는 오늘까지 문제를 내야 한다는 사실을 깨달았다. 그러나 게임은 시작되었고 지는 걸 무척이나 싫어하는 유섭이는 어쩔 수 없이 백도어를 해 게임을 최대한 빠르게 끝내기로 결심하였다.

최대한 빨리 게임을 끝내고 문제를 출제해야 하기 때문에 유섭이는 최대한 빨리 넥서스가 있는 곳으로 달려가려고 한다. 유섭이의 챔피언은 총 N 개의 분기점에 위치할 수 있다. 0번째 분기점은 현재 유섭이의 챔피언이 있는 곳을, $N - 1$ 번째 분기점은 상대편 넥서스를 의미하며 나머지 $1, 2, \dots, N - 2$ 번째 분기점은 중간 거점들이다. 그러나 유섭이의 챔피언이 모든 분기점을 지나칠 수 있는 것은 아니다. 백도어의 핵심은 안 들키고 살금살금 가는 것이기 때문에 적 챔피언 혹은 적 와드(시야를 밝혀주는 토탈), 미니언, 포탑 등 상대의 시야에 걸리는 곳은 지나칠 수 없다.

입력으로 각 분기점을 지나칠 수 있는지에 대한 여부와 각 분기점에서 다른 분기점으로 가는데 걸리는 시간이 주어졌을 때, 유섭이가 현재 위치에서 넥서스까지 갈 수 있는 최소 시간을 구하여라.

입력 형식

첫 번째 줄에 분기점의 수와 분기점들을 잇는 길의 수를 의미하는 두 자연수 N 과 M 이 공백으로 구분되어 주어진다. ($2 \leq N \leq 100,000$, $1 \leq M \leq 300,000$)

두 번째 줄에 각 분기점이 적의 시야에 보이는지를 의미하는 N 개의 정수 a_0, a_1, \dots, a_{N-1} 가 공백으로 구분되어 주어진다. a_i 가 0이면 i 번째 분기점이 상대의 시야에 보이지 않는다는 뜻이며, 1이면 보인다는 뜻이다. 추가적으로 $a_0 = 0$, $a_{N-1} = 1$ 이다. $N - 1$ 번째 분기점은 상대 넥서스이기 때문에 어쩔 수 없이 상대의 시야에 보이게 되며, 또 유일하게 상대 시야에 보이면서 갈 수 있는 곳이다.

다음 M 개의 줄에 걸쳐 세 정수 a, b, t 가 공백으로 구분되어 주어진다. ($0 \leq a, b < N$, $a \neq b$, $1 \leq t \leq 100,000$) 이는 a 번째 분기점과 b 번째 분기점 사이를 지나는데 t 만큼의 시간이 걸리는 것을 의미한다. 연결은 양방향이며, 한 분기점에서 다른 분기점으로 가는 간선은 최대 1개 존재한다.

출력 형식

첫 번째 줄에 유섭이의 챔피언이 상대 넥서스까지 안 들키고 가는데 걸리는 최소 시간을 출력하여라. 만약 상대 넥서스까지 갈 수 없으면 -1 을 출력하여라.

예제

표준 입력	표준 출력
5 7 0 0 0 1 1 0 1 7 0 2 2 1 2 4 1 3 3 1 4 6 2 3 2 3 4 1	12
5 7 0 1 0 1 1 0 1 7 0 2 2 1 2 4 1 3 3 1 4 6 2 3 2 3 4 1	-1

설명

첫 번째 예제에서 위 그래프의 최단거리는 $0 \rightarrow 2 \rightarrow 3 \rightarrow 4$ 를 지나는 시간인 $5(2+2+1)$ 이지만, 3번 분기점이 상대의 시야에 있기 때문에 $0 \rightarrow 2 \rightarrow 1 \rightarrow 4$ 를 지나는 시간인 $12(2+4+6)$ 이 최소 시간이 된다.

문제 J. FLEX

시간 제한: 2 초

메모리 제한: 512 MB

요즘 스트레스를 많이 받는 인호에게 돈은 스트레스 해소 그 자체이다. 그 때문인지 인호는 어제 돈을 썼던만큼 오늘 쓰지 못하면, 만 원 단위로 표현한 차액의 제곱만큼 박탈감을 느낀다. 보다 엄밀히 서술하자면 다음과 같다.

만약 i 번째 날 지출이 a 만 원, $(i + 1)$ 번째 날 지출이 b 만 원이었다면, $(i + 1)$ 번째 날에

1. $a \leq b$ 라면 아무런 박탈감을 느끼지 않는다.
2. $a > b$ 라면 $(a - b)^2$ 만큼의 박탈감을 느낀다.

인호는 앞으로 N 일 동안의 예상 통장 지출 내역을 살펴보고 있다. 인호의 통장에는 예상 지출과 별개로 여유롭게 사용할 수 있는 M 만 원의 돈이 남아있다. 인호는 이 M 만 원을 앞으로의 지출 내역에 효율적으로 분배하여, 앞으로 N 일 동안 느낄 박탈감의 합(즉, 내일의 박탈감, 모레의 박탈감, ..., N 일 후의 박탈감의 합)을 최소화하고자 한다. 이 통장 잔고의 최소 인출 단위는 만 원이기 때문에, 분배는 만 원 단위로만 할 수 있다. 슬하고 더운 날씨 속에서 인호가 박탈감을 덜 받을 수 있게 도와주자.

입력 형식

첫 번째 줄에는 정산할 날짜의 수와 통장 잔고(만 원 단위)를 의미하는 두 자연수 N 과 M 이 공백으로 구분되어 주어진다. ($1 \leq N < 1000$, $0 \leq M < 200$)

두 번째 줄에는 N 일 동안 지출할 예상 비용을 의미하는 N 개의 정수 C_1, C_2, \dots, C_N 가 공백으로 구분되어 주어진다. ($0 \leq C_i \leq 10$)

C_i 는 i 번째 날에 지출할 비용이 C_i 만 원임을 의미한다. 0번째 날에 지출한 비용은 0으로 간주한다.

출력 형식

첫 번째 줄에 통장 잔고를 N 일간의 예상 지출액에 분배 하여 인호가 앞으로의 N 일 동안 느낄 박탈감의 총합의 최솟값을 출력한다.

예제

표준 입력	표준 출력
3 1	1
3 2 1	

설명

인호는 3일동안 각각 3, 2, 1만 원을 지출할 예정이고, 추가로 사용 가능한 잔액 1만 원이 남아있다.

원래는 총 2만큼의 박탈감을 느껴야 하지만 마지막 날에 1만 원을 추가한다면, 예상 지출 비용이 [3, 2, 2]가 되어 1만큼의 박탈감만을 느끼게 해줄 수 있다 (둘째 날에만 1을 느낀다).

3일 동안만의 박탈감을 따지는 것이기 때문에 4번째 날의 비용을 상정할 필요가 없음을 유의해야 한다.

이와 별개로, 기존의 예상 지출 자체를 재분배할 수는 없다. 예를 들어 [3, 2, 1]을 [2, 2, 2]로 바꾸는 것은 허용되지 않는다.

참고 사항

flex는 본디 구부리다, (준비 운동으로) 몸을 풀다의 뜻을 가지고 있는 영단어인데, 운동하며 나오는 근육을 과시함을 뜻하는 관용구 'flex your muscles'로도 쓰인다.

최근에는 몸풀기로 근육을 자랑하는 게 아니라 돈 등의 물질적인 자원을 자랑하는 은어로 사용되고 있다.

문제 K. 통신망 분할

시간 제한: 1 초

메모리 제한: 512 MB

BOJ의 인기스타, 방송인 권옥제는 통신 회사에 취업했다. 현재 이 통신 회사는 너무나 큰 통신망을 한 지사에서 관리하느라 큰 비용을 지불하고 있었다. 그래서 회사는 최근 IT의 트렌드 중 하나인 ‘탈중앙화’에 편승하여, 통신망을 분할하도록 결정했다. 그래서 옥제한테 통신망을 분할 할때 발생하는 비용을 분석하도록 지시했다.

현재 회사 망에는 1번부터 N 번까지 총 N 개의 통신 탑이 존재하며, 통신탑 간의 연결이 M 개 존재한다. 이때 회사에서는 총 Q 번 통신탑 간의 연결을 제거함으로써 하나의 통신망을 여러 개의 통신망으로 분리하려고 한다. 통신망이란, 통신탑의 연결을 통해 도달 가능한 통신탑들의 집합이다. 통신탑 간의 연결 관계를 제거할 때 드는 비용은 제거한 후 통신망이 두 개로 나누어진다면 나뉜 두 개의 통신망에 속한 통신탑들의 갯수의 곱이 되며, 나누어지지 않을 경우 0이다.

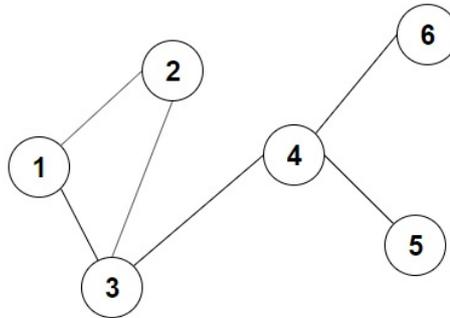


Figure 2: 예시 그래프

위의 그림을 예시로 할때, (3,4) 간의 연결을 제거하면 {1, 2, 3}, {4, 5, 6}으로 분할 되며, 이때 발생하는 비용은 $3 \times 3 = 9$ 가 된다. 대신 (2,3) 간의 연결을 제거하면, 망이 나뉘지지 않았기에 비용은 0이 된다.

옥제는 회사의 요청에 따라 Q 번의 제거를 통해 나오는 비용의 합을 구해야 한다. 하지만 옥제는 회사의 통신망을 이용해 방송하기 바쁘기 때문에 우리가 도와주자.

입력 형식

첫 번째 줄에는 통신탑의 개수인 자연수 N , 통신탑 사이의 연결의 개수인 자연수 M , 통신망 연결 분할 횟수인 자연수 Q 가 공백으로 구분되어 주어진다. ($1 \leq N \leq 100,000$, $1 \leq M \leq 100,000$, $1 \leq Q \leq M$)

두 번째 줄부터 M 개의 줄에 걸쳐 두 개의 자연수 X, Y 가 공백으로 구분되어 주어진다. 이는 X 통신탑과 Y 통신탑 사이에 연결이 있음을 뜻한다. ($1 \leq X, Y \leq N$, $X \neq Y$)

중복된 연결은 주어지지 않으며, 모든 통신탑은 처음엔 하나의 통신망에 속한다. 조건에 의해 자기 자신과 연결이 있는 통신탑은 없다.

그 다음 줄부터 Q 개의 줄에 걸쳐 제거될 연결의 번호인 자연수 A 가 주어진다. 이는 A 번째로 입력된 (X, Y)의 연결이 제거되었음을 의미한다. ($1 \leq A \leq M$)

이미 제거된 연결은 다시 제거되지 않으며, 제거는 입력 순서대로 진행된다.

출력 형식

첫 번째 줄에 Q 개의 연결을 순서대로 제거하는데 드는 비용의 합을 출력한다.

예제

표준 입력	표준 출력
4 4 3 1 2 2 3 3 4 1 4 4 2 3	5

설명

첫 번째로 제거되는 연결은 (1,4)로, 통신망 {1, 2, 3, 4}가 분리되지 않아 비용이 0이다.

두 번째로 제거되는 연결은 (2,3)으로, 통신망 {1, 2, 3, 4}가 {1, 2}와 {3, 4}로 분리되어 비용이 $2 \times 2 = 4$ 이다.

세 번째로 제거되는 연결은 (3,4)로, 통신망 {3, 4}가 {3}과 {4}로 분리되어 비용이 $1 \times 1 = 1$ 이다.

결과적으로 총 비용은 $0 + 4 + 1 = 5$ 이다.

문제 L. 깃발춤

시간 제한: 2 초

메모리 제한: 512 MB

매일 코딩만 하던 상현이는 두뇌가 너무 코드에만 범벅이 되고 있는 것 같아 두뇌를 다른 방면으로 몰입시킬 수 있는 깃발춤 공연을 보러 가기로 했다. 깃발춤 공연은 N 명의 공연자들이 일렬로 서서 깃발을 박력 있게 흔들며 진행된다. 깃발을 들고 있는 공연자들은 각각 카리스마의 정도 c_i 를 지니고 있어, 몇몇 공연자들은 보다 절도 있게 깃발을 흔든다.

상현이는 깃발춤을 보다가, 문득 깃발춤 공연 중 몇몇 연속된 공연자들이 깃발을 교대로 흔드는 것을 목격했다. 상현이는 이를 ‘교대 깃발춤’이라 명명하였다. 교대 깃발춤은 L 번째 공연자부터 R 번째 공연자까지 깃발을 각자 왼쪽 또는 오른쪽으로 흔드는 동작인데, L 번째 공연자를 포함하여 L 번째 공연자와의 거리가 짝수인 공연자는 깃발을 왼쪽으로 흔들고, 거리가 홀수인 공연자는 오른쪽으로 흔든 뒤, 다시 깃발을 몸 쪽으로 원위치시킨다. 여기서 x 번째 공연자와 y 번째 공연자 사이의 거리는 $|x - y|$ 로 표현된다.

문제 해결에서 벗어날 수 없었던 상현이는 교대 깃발춤에서 왼쪽으로 깃발을 흔든 공연자들의 카리스마의 합과 오른쪽으로 깃발을 흔든 공연자들의 카리스마의 합의 차이의 절댓값을 교대 깃발춤의 균일도라고 부르기로 하였다. 교대 깃발춤의 균일도가 큰 값을 가지면, 한쪽이 다른 쪽보다 압도적으로 카리스마가 느껴진다는 뜻이기에 비대칭적으로 보일 수 있다. 상현이는 교대 깃발춤의 균일도가 중요한 의미를 지닌다고 생각한다. 또 깃발을 흔드는 공연자들은 공연의 열기와 순간순간의 실수에 휩쓸리기 때문에, 공연자들의 카리스마가 증가하거나 감소할 수 있다. 이런 모든 상황을 고려하며 상현이는 매 교대 깃발춤의 균일도를 구하고 싶어졌다. 상현이를 도와주자!

입력 형식

첫 번째 줄에는 깃발춤을 진행하는 공연자의 명수인 자연수 N 과 상황 변화의 개수인 자연수 Q 가 공백으로 구분되어 주어진다. ($1 \leq N \leq 300,000, 1 \leq Q \leq 300,000$)

두 번째 줄에는 정수 c_1, c_2, \dots, c_N 이 공백으로 구분되어 주어지며, c_i 는 i 번째 공연자의 카리스마를 의미한다. ($-100,000 \leq c_i \leq 100,000$)

세 번째 줄부터 Q 개의 줄에 걸쳐 다음 형식 중 하나로 세 정수가 공백으로 구분되어 주어진다.

- $1 L R$: L 번째 공연자부터 R 번째 공연자까지 구성된 교대 깃발춤이 시연된다. ($1 \leq L \leq R \leq N$)
- $2 L x$: L 번째 공연자의 카리스마가 정수 x 만큼 증가한다. ($1 \leq L \leq N, -100,000 \leq x \leq 100,000$)

첫 번째 종류(‘ $1 L R$ ’ 꼴)의 쿼리는 한 번 이상 주어짐이 보장된다.

출력 형식

첫 번째 종류의 쿼리가 입력될 때마다 매 줄에 해당하는 교대 깃발춤의 균일도를 출력한다.

예제

표준 입력	표준 출력
6 3	2
3 1 4 1 5 9	9
1 2 4	
2 3 10	
1 3 6	

설명

첫 번째 쿼리에서 교대 깃발춤을 추는 공연자들의 카리스마는 순서대로 1, 4, 1이다. 균일도는 $|(1 + 1) - (4)| = 2$ 이다.

두 번째 쿼리로 인해 3번째 공연자의 카리스마가 10 증가하여, 전체 카리스마가 3, 1, 14, 1, 5, 9가 된다.

세 번째 쿼리에서 교대 깃발춤을 추는 공연자들의 카리스마는 순서대로 14, 1, 5, 9이다. 균일도는 $|(14 + 5) - (1 + 9)| = 9$ 이다.

문제 M. 일하는 세포

시간 제한: 1 초

메모리 제한: 512 MB

사람의 세포 수, 약 37조 개. 세포들은 몸 속에서 오늘도 열심히 일하고 있다. 그중에서도 우리의 적혈구는 혈관을 타고 돌아다니며 산소와 영양소를 운반해주는 중요한 역할을 맡고 있다.

적혈구는 심장이나 폐 같은 거점들을 돌아다니면서 산소와 영양분을 운반한다. 몸 속에는 총 N 개의 거점이 있고, 몇몇 거점은 통로를 통해 서로 이어져 있다. 거점 사이의 통로를 통과하는데는 1초가 걸린다. 하지만 혈관의 곳곳에는 판막이나 공사중인 부분들이 있기 때문에 때 초 거점 사이의 연결관계가 바뀐다. 그럼에도 불구하고 몸의 곳곳이 산소와 영양분을 필요로 하므로 적혈구는 가만히 있을 수 없으며, 때 초 통로를 무조건 하나 타야 한다. 일부 통로는 출발 거점과 도착 거점이 같을 수도 있다. 일부 거점의 특정 순간에는 나가는 통로가 없을 수도 있는데, 이 때는 도착한지 1초 후에 파괴되어 몸과 다시 하나가 된다. 잔혹하지만 우리의 몸은 이렇게 돌아간다.

우리의 적혈구는 매 순간 변하는 몸속 혈관 지도에 길을 헤매지만 그래도 최선을 다해서 하루하루 열심히 일하고 있다. 옆에 있던 백혈구가 길을 헤매는 적혈구를 보고 도와주고 싶다는 생각을 했다.

수십 시간의 유주 경험을 통해 백혈구는 몸속 혈관 지도가 T 초를 주기로 반복된다는 것을 알게 되었다. 이 사실을 정리해서 적혈구가 거점 A 에서 출발하여 정확히 D 초 후 거점 B 에 도달하게 되는 경우의 수를 모든 거점의 순서쌍에 대해 구해주고자 하지만 너무나도 단세포이기 때문에 머리가 나빠서 계산을 하지 못했다. 한 경로는, D 초 동안 통과한 통로의 순열로 정의된다. 백혈구를 도와서 적혈구가 D 초 동안 한 거점에서 다른 거점까지 움직일 수 있는 경우의 수를 구해주자!

입력 형식

첫 번째 줄에는 백혈구가 알아낸 혈관 지도들의 주기인 자연수 T 와 거점의 개수인 자연수 N , 적혈구가 움직이는 시간인 정수 D 가 공백으로 구분되어 주어진다. ($1 \leq T \leq 100$, $2 \leq N \leq 20$, $0 \leq D \leq 10^9$)

그 뒤 거점 사이의 연결 관계를 나타내는 혈관 지도 T 개가 순서대로 1번부터 T 번까지 주어지는데, 혈관 지도가 주어지는 형식은 다음과 같다.

- 첫 번째 줄에는 거점 사이를 잇는 혈관의 개수인 자연수 M_i 가 주어진다. ($0 \leq M_i \leq N^2$)
- 그 뒤 M_i 개의 줄에 걸쳐 세 자연수 a, b, c 가 공백으로 구분되어 주어진다. 이는 거점 a 에서 거점 b 로 가는 서로 다른 단방향 통로가 c 개 있음을 의미한다. ($1 \leq a, b \leq N$, $1 \leq c \leq 1000$)

매 혈관 지도에 중복된 연결 관계는 주어지지 않으며, i 초에서 $(i+1)$ 초 동안 이동할 때는 $(i \% T + 1)$ 번 혈관 지도가 적용된다. $i \% T$ 는 i 를 T 로 나눈 나머지를 의미한다.

출력 형식

출력은 N 개의 줄로 구성되며, i 번째 줄에는 N 개의 정수 $x_{i1}, x_{i2}, \dots, x_{iN}$ 를 공백으로 구분하여 출력해야 한다. x_{ij} 는 0초 때 거점 i 에서 출발하여 정확히 D 초 때 거점 j 에 위치하게 되는 경로의 수를 1,000,000,007로 나눈 나머지이다.

예제

표준 입력	표준 출력
1 2 4 2 1 1 2 2 2 3	16 0 0 81
3 4 5 4 1 2 1 2 3 1 3 4 1 4 1 1 4 1 1 2 2 3 1 3 4 1 4 2 1 5 1 2 1 2 4 1 4 3 1 3 1 1 1 4 1	0 0 1 0 0 1 0 0 2 0 0 0 4 0 0 2
1 8 3 12 1 2 1 1 8 1 2 3 1 2 8 1 3 4 1 3 7 1 3 8 1 4 5 1 4 7 1 5 6 1 6 7 1 7 8 1	0 0 0 1 0 0 1 1 0 0 0 0 1 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

표준 입력	표준 출력
1 8 100000000	261245548 769313318 167840464
24	862450688 445583828 270525651
1 2 1	828293276 976953408
1 8 1	769313318 542054741 65223362
2 1 1	957807341 63263610 545566670
2 3 1	134857214 863984679
2 8 1	167840464 65223362 959076197
3 2 1	916983285 988077461 199284746
3 4 1	461375786 371787307
3 7 1	862450688 957807341 916983285
3 8 1	119640157 267995930 978327505
4 3 1	847171719 483910227
4 5 1	445583828 63263610 988077461
4 7 1	267995930 394594439 718634258
5 4 1	715295769 69712722
5 6 1	270525651 545566670 199284746
6 5 1	978327505 718634258 131562703
6 7 1	197248645 728310434
7 3 1	828293276 134857214 461375786
7 4 1	847171719 715295769 197248645
7 6 1	322189612 142912983
7 8 1	976953408 863984679 371787307
8 1 1	483910227 69712722 728310434
8 2 1	142912983 162707920
8 3 1	
8 7 1	

설명

네 번째 예제의 경우 적혈구가 3.17년 동안 몸 속을 돌아다니는 경우이다. 표의 너비가 좁아 8개의 숫자들이 한 줄에 있어야 하나 3개 / 3개 / 2개로 나뉘어 표시되었다.

문제 N. 트리의 외심

시간 제한: 2 초

메모리 제한: 512 MB

알고리즘에 푼 빠진 동판이가 트리에 심취한 나머지 트리에서 외심을 정의하려 한다. 트리란, 모든 정점이 연결되어 있으면서 사이클이 존재하지 않는 그래프이다. 하지만 동판이는 트리에서 외심을 정의하기 위해서는 “트리에서 두 정점 사이의 거리”도 정의해야 한다는 사실을 깨달았다!

트리에서 두 정점 사이의 거리는 한 정점에서 다른 정점으로 가기 위해 거쳐야 하는 최소한의 간선의 개수로 정의된다. 이 때 트리의 세 정점에 대해, 트리의 외심은 세 정점으로부터 거리가 같으면서, 그 거리를 최소화 하는 정점이 존재한다면 해당 정점으로 정의된다. 수학적으로 트리의 세 정점에 대해 외심이 존재한다면, 유일하다는 것을 보일 수 있다.

자명하게도, 외심을 정의하는 3개의 정점이 달라지면 같은 트리라 해도 외심이 달라진다. 동판이는 다양한 외심들을 찾아보고 싶지만 코딩에 귀찮음을 겪고 있다.....동판이를 위해 여러분들이 대신 코드를 짜주도록 하자.

입력 형식

첫 번째 줄에 정점의 개수 N 이 주어진다. ($1 \leq N \leq 100,000$) 이 트리는 1번 정점, 2번 정점, ..., N 번 정점으로 구성된다.

두 번째 줄부터 N 번째 줄까지, 트리의 간선 정보를 의미하는 두 자연수 X, Y 가 공백으로 구분되어 주어진다. 이는 X 번 정점과 Y 번 정점이 연결되어있음을 의미한다. ($1 \leq X, Y \leq N, X \neq Y$)

주어지는 연결관계는 트리를 구성한다.

$N + 1$ 번째 줄에는 쿼리의 개수 Q 가 주어진다. ($1 \leq Q \leq 100,000$)

다음 Q 개의 줄에 걸쳐, 외심을 정의하기 위한 세 개의 정점 번호를 뜻하는 세 자연수 A, B, C 가 공백으로 구분되어 주어진다. ($1 \leq A, B, C \leq N$)

출력 형식

Q 개의 줄에 걸쳐 각 쿼리마다 입력으로 주어진 세 정점에 대해 트리의 외심이 존재하면 외심의 정점 번호를, 존재하지 않으면 -1 을 출력한다.

예제

표준 입력	표준 출력
4 1 2 1 3 1 4 2 2 3 4 1 2 3	1 -1
2 1 2 2 1 1 2 2 2 2	-1 2
6 1 2 2 3 2 4 3 5 5 6 1 1 4 6	3

문제 O. 시간 끌기

시간 제한: 1 초

메모리 제한: 512 MB

영희와 철수는 공부하기에 앞서 게임을 하고 있다. 게임은 $N \times M$ 크기의 게임판에서 진행되는데 몇몇 칸은 X 표시가 있다.

게임은 두 명이 번갈아 가며 진행되고 다음의 규칙을 따른다.

1. 아직 선택되지 않은 행 또는 열을 하나 선택하되, 기존에 선택되었던 행과 열과의 교차점에 X가 있어서는 안 된다.
2. 선택한 후, 상대의 차례가 된다.
3. 만약 조건을 만족하는 행 또는 열이 없다면 (또는 이미 모든 행 또는 열을 선택했다면) 그 사람은 지고 게임은 끝난다.

예를 들어 다음의 게임판을 보자.

X		X	
			X
	X		X
		X	

지금까지 두 번째 행과 네 번째 행, 두 번째 열을 선택한 상태라고 하자. 그렇다면 첫 번째 행을 선택하거나 네 번째 열을 선택하는 것 외에는 선택된 행, 열들의 교차점에 X가 생기게 된다. 따라서 첫 번째 행이나 첫 번째 열을 선택해야만 하고 그 뒤에는 어떤 행이나 열을 골라도 교차점에 X가 생기므로 총 선택 횟수는 4번이며, 이보다 더 많이 선택할 수 있는 방법은 없다.

철수와 영희는 공부가 매우 하기 싫었기에 누가 이기는지와는 별개로 가장 오래가는 선택을 한다고 가정하자. 이때 철수와 영희는 도합 최대 몇 번의 선택을 할 수 있는지를 출력하시오.

입력 형식

첫 번째 줄에 게임판의 사이즈인 N 과 M , 그리고 X 표시된 칸의 개수 K 가 공백으로 구분되어 주어진다. ($1 \leq N, M \leq 200, 1 \leq K \leq NM$)

이후 K 개의 줄에 걸쳐 두 자연수 x, y 가 공백으로 구분되어 주어진다. ($1 \leq x \leq N, 1 \leq y \leq M$) 이는 x 행 y 열에 있는 칸에 X 표시가 있다는 것을 의미한다.

X 표시가 있는 칸이 중복되어서 입력되는 경우는 없다.

출력 형식

첫 번째 줄에 철수와 영희가 합쳐서 총 몇 번의 선택을 하게 되는지 출력한다.

예제

표준 입력	표준 출력
4 4 6 1 1 1 3 2 4 3 2 3 4 4 3	4

문제 P. 가장 높고 넓은 성

시간 제한: 1 초

메모리 제한: 512 MB

당신은 어마어마한 크기의 토지를 소유한 부자다. 평지에는 n 개의 표지판이 있다. 당신은 지금부터 이 곳에 민성이를 위한 성을 지을 것이다. 성은 여러 개의 층으로 구성된 구조물이다.

- 1층은 평지 위에 지어지고, 2층은 1층 위에, 3층은 2층 위에, \dots , n 층은 $n - 1$ 층 위에 지어진다.
- 각 층의 경계는 다각형이며, 각 층의 경계의 꼭짓점에는 표지판이 위치하여야 한다. 다시 말하면, 표지판 위에만 각 층의 경계의 꼭짓점을 세울 수 있다.
- 한 표지판은 여러 층의 경계의 꼭짓점에 동시에 사용될 수 없다. 그러나 한 표지판이 다른 층의 꼭짓점이 아닌 경계에 위치할 수는 있다.
- 넓이가 0인 층은 허용되지 않는다.
- 쓸모가 없는 표지판들은 버려도 된다.

민성이는 숙소를 고를 때 전망을 최우선으로 고려하므로, 당신이 성을 지을 때 고려해야 할 우선순위는 다음과 같다.

1. 성의 높이를 가능한 한 제일 높게 지어야 한다.
2. 성의 모든 층의 넓이의 합이 가능한 한 제일 크게 지어야 한다.
3. 가능한 적은 수의 표지판을 사용하여야 한다.

당신은 성을 몇 층까지 지을 수 있으며, 그 때 각 층에 사용될 표지판들이 무슨 표지판인지 알아내야 한다.

입력 형식

첫 번째 줄에 표지판의 개수 n 이 주어진다. ($1 \leq n \leq 10^3$)

두 번째 줄부터 n 개의 줄에 걸쳐 각 표지판들의 위치를 의미하는 정수 x, y 가 공백으로 구분되어 주어진다. ($-10^4 \leq x, y \leq 10^4$) 이는 (x, y) 에 표지판이 위치함을 의미한다.

모든 표지판은 서로 다른 위치에 세워져 있다.

출력 형식

첫 번째 줄에 n 개의 정수 x_1, x_2, \dots, x_n 를 공백으로 구분하여 출력한다. x_i 는 i 번째 표지판이 사용되었을 경우 사용된 층수이며, 사용되지 않았으면 0이다.

