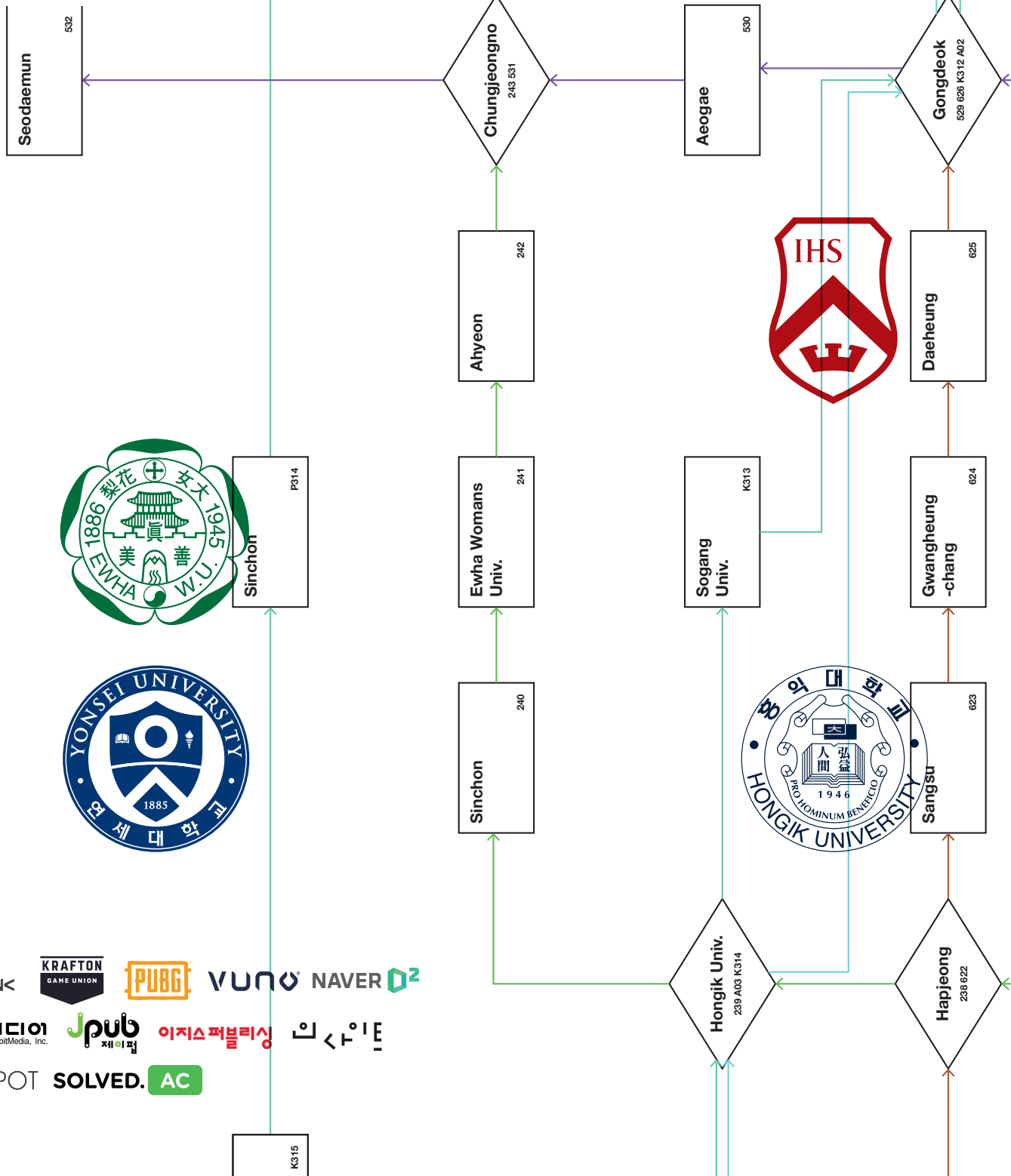


2020 신촌지역
대학생 프로그래밍 대회 동아리
연합 여름 대회

SUAPC 2020

Division 1

Official Problemset



문제 목록

문제지에 있는 문제가 총 12문제가 맞는지 확인하시기 바랍니다.

- A** 200년간 폐관수련했더니 PS 최강자가 된 건에 대하여
- B** 꿀벌
- C** 케이크 커팅
- D** 가뭄 (Large)
- E** 독특한 계산기
- F** 울타리
- G** 마스크가 필요해
- H** 전설
- I** 수학은 재밌어
- J** 상품권 준비
- K** 물건 가져가기
- L** 카드 셔플

모든 문제의 메모리 제한은 1GB로 동일합니다.

문제 A. 200년간 폐관수련했더니 PS 최강자가 된 건에 대하여

시간 제한 1 초
메모리 제한 1024 MB

2220년에도 “2220 신춘지역 대학생 프로그래밍 대회 동아리 연합 수시 대회”가 성공적으로 개최된다. SUAPC은 이제 모든 학생이 즐길 수 있도록 다양한 난이도의 대회가 1년에 수시로 열리며, 상금 규모는 역대가 되었다. 이런 미래를 예측한 연두는 냉동인간이 되어 폐관수련을 통해 알고리즘만 공부하였으며, 이제 모든 대회에서 반드시 1등을 한다.

하지만 각 대회마다 참가자격이 생겨 모든 대회에 참가하지 못할 수도 있다! 상금의 독식을 막기 위해 대회마다 “상금 상한”이 존재하며, 어떤 대회를 참가하기 전까지 모은 상금의 합이 그 대회의 상금 상한을 초과한다면 그 대회는 참가할 수 없다. 대회가 열리는 순서는 정해져 있고 대회들의 시간은 겹치지 않는다.

올해 열리는 N 개의 대회에 모두 참가하여 상금을 쓸어 모을 계획을 세웠던 연두는 이 사실을 알고 큰 충격을 받았다. 대회를 하나까지 참가하지 못하는 것은 참을 수 있지만, 2개 이상의 대회를 참가할 수 없다면 절망에 빠져 500년을 더 냉동인간 상태로 지낼 것이다. 아직 손이 덜 해동된 연두를 위하여, 올해 연두가 적어도 $N - 1$ 개의 대회에 참가할 수 있는지 여러분이 대신 확인해주자.

입력

첫 번째 줄에 올해 열리는 대회의 수 N 이 주어진다. ($1 \leq N \leq 100,000$)

다음 N 개의 줄에 i 번째 대회에 대한 정보인 정수 x_i, p_i 가 대회가 열리는 순서대로 주어진다. ($1 \leq x_i, p_i \leq 10^9$). 이는 i 번째 대회에 참가하기 전까지 모은 상금의 합이 x_i 원 이하여야 이 대회에 참가할 수 있고, 이 대회에 참가하면 연두가 얻는 상금이 p_i 원임을 의미한다.

출력

만약 올해 연두가 최소 $N - 1$ 개의 대회에 참가할 수 있으면 Kkeo-eok을, 참가할 수 없으면 Zzz를 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
4 10000 1000 2000 5000 3000 4000 5000 10000	Kkeo-eok

두 번째 대회를 제외하면 모든 대회에 참가할 수 있다.

표준 입력(stdin)	표준 출력(stdout)
4 10000 5000 10000 5000 4000 1000 4000 1000	Zzz

이 페이지는 공백입니다

문제 B. 꿀벌

시간 제한 1 초
메모리 제한 1024 MB

작은 벌집에 살고 있는 꿀벌이 있다. 오늘은 벌집 여기저기를 돌아다니면서 벌집에 저장되어 있는 꿀을 배불리 먹으려 한다.

벌집의 각 칸에는 꿀이 저장되어 있는데, 꿀을 먹어서 얻거나 잃는 에너지는 칸마다 다르다. 꿀벌은 아무 칸에서나 시작해서 다음과 같이 움직이면서 방문하는 모든 칸에 있는 꿀을 전부 먹고 아무 칸에서나 나올 것이다. 한 칸도 방문하지 않는 것도 가능하다.

- 인접한 칸으로 이동한다. 이동하는 데에는 에너지가 소모되지 않는다.
- 벌집 안의 다른 임의의 칸으로 날아간다. 벌의 눈은 육각형 모양이기 때문에, 항상 벌집의 벽과 수직인 6개 방향 중 하나로 날아다닌다. 날아다니는 동안 방향을 바꿀 수 있다. 날아가는 데에는 $(\text{벌집 거리} - 1) \times F$ 만큼의 에너지가 소모된다.
 - 어느 두 칸의 **벌집 거리**는, 인접한 칸의 거리를 1이라고 정의하고, 어떤 한 칸에서 다른 한 칸으로 인접한 칸들만 거쳐서 이동했을 때의 두 칸의 최단 경로의 거리로 정의한다.
- 벌은 날아다니면서 페로몬을 뿌린다. 페로몬은 피로를 없애 주기 때문에, 방향과 상관없이 날아왔던 경로를 다시 지나는 경우에는 에너지가 소모되지 않는다. 또한 페로몬은 금방 사라지지 않아서 나중에 다시 같은 경로를 지나더라도 에너지를 소모하지 않는다.

벌집의 크기 N 과, 모든 칸에 대해 각 칸의 꿀을 먹었을 때 얻거나 잃는 에너지의 정보가 주어질 때, 꿀벌이 얻을 수 있는 최대 에너지를 계산하자. 벌집의 크기가 N 이라는 것은 벌집의 가운데 칸에서 벌집 거리로 가장 멀리 갈 수 있는 칸까지의 벌집 거리가 N 이라는 뜻이다.

입력

첫째 줄에는 벌집의 크기 N 과 문제에서 언급한 상수 F 가 주어진다. ($1 \leq N \leq 6, 1 \leq F \leq 10^9$)

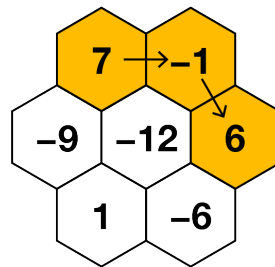
두 번째 줄부터 $(2N - 1)$ 개 줄에는 각 칸마다 벌집에 저장되어 있는 꿀을 먹어서 얻을 수 있는 에너지의 정도가 위에서 아래, 왼쪽에서 오른쪽 순서대로 주어진다. 에너지는 절댓값이 10^9 보다 크지 않은 정수이다.

출력

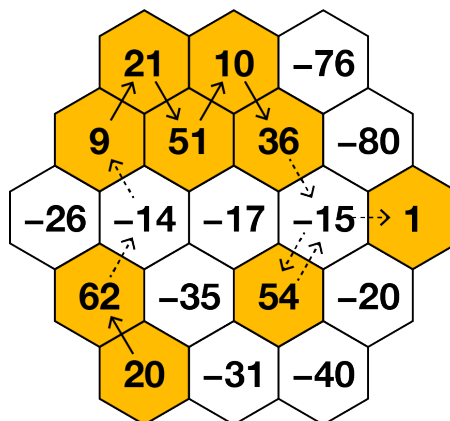
꿀벌이 얻을 수 있는 최대 에너지를 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
2 5 7 -1 -9 -12 6 1 -6	12



표준 입력(stdin)	표준 출력(stdout)
3 10 21 10 -76 9 51 36 -80 -26 -14 -17 -15 1 62 -35 54 -20 20 -31 -40	244



-15에서 54로 날아갈 때는 에너지 10이 소모되지만, 54에서 -15로 돌아올 때는 에너지가 소모되지 않는다.

문제 C. 케이크 커팅

시간 제한 1 초
메모리 제한 1024 MB

원기둥 모양의 치즈 케이크가 있다. 치즈 케이크의 중심의 좌표는 $(0, 0)$ 이다.

생쥐들이 케이크의 일부를 먹었다. 생쥐들은 원기둥 모양으로 바닥까지 모두 파먹었고, 생쥐들이 먹은 부분은 서로 겹치지 않는다.

케이크의 한 부분은 페퍼민트로 장식되어 있다. 생쥐는 페퍼민트 향을 싫어하기 때문에 그 부분은 생쥐가 먹지 않는다.

밀면에 수직인 평면 모양으로 칼질을 하려고 한다. 칼질을 정확히 한 번만 해서 페퍼민트로 되어 있는 부분을 지나고 케이크의 양이 정확히 반이 되도록 나누어라.

입력

첫번째 줄에는 케이크의 반지름 R ($1 \leq R \leq 1,000$)과 페퍼민트의 좌표 x_p, y_p 가 주어진다. R 과 페퍼민트의 좌표는 정수이고, 페퍼민트가 케이크 안에 있음이 보장된다.

다음 줄에는 생쥐의 수 N ($1 \leq N \leq 100$)이 주어진다.

다음 N 줄에는 각 생쥐가 먹은 케이크의 중심 좌표 x_i, y_i 와 반지름 r_i 가 주어진다. 모든 좌표와 반지름은 정수이고, 원기둥은 케이크에 완전히 포함되며 서로 겹치지 않는다. 페퍼민트는 생쥐가 파먹은 어떤 원기둥에도 포함되지 않는다.

출력

칼질한 평면을 위에서 바라보면 직선 모양이 된다. 이 직선의 각도를 x 축에서 반시계 방향으로 재어 라디안 단위로 출력한다. 출력한 각도대로 잘랐을 때 한 부분의 넓이를 A , 전체 넓이의 절반을 B 라고 할 때, A 와 B 의 절대/상대 오차가 10^{-6} 이하이면 정답이다. 출력값의 범위는 지정되어 있지 않지만, 정확한 채점을 위해 -2π 이상 2π 이하의 값을 출력하는 것을 권장한다.

만약 어떤 방식으로든 정확히 절반을 자를 수 없다면 **NO**를 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
100 -50 0 1 0 0 10	0.00000000000
10 3 -4 2 4 3 5 -4 -3 5	2.21429743558818

표준 입력(stdin)	표준 출력(stdout)
10 0 0 2 4 3 5 -4 -3 5	0.123456789101112

어떤 각도로 자르더라도 항상 절반으로 나뉜다.

표준 입력(stdin)	표준 출력(stdout)
1000 278 960 1 -1 0 999	1.987912112427

문제 D. 가뭄 (Large)

시간 제한 1 초
메모리 제한 1024 MB

가뭄에 찌든 신촌을 위해서 국렬이는 신촌에 비를 내렸다. 가뭄(Small)에서 충분히 비를 내리게 해서 홍익대학교와 이화여자대학교는 가뭄을 해결하였다. 그러나 신촌역 근방이라 제일 크게 피해를 본 연세대학교와 서강대학교는 가뭄이 완전히 해결되지 않았기 때문에 국렬이는 이 두 대학교에 다시 비를 내리게 할 것이다.

각 대학교의 구역들은 N 칸으로 구분되어 있으며 연세대학교의 구역은 $A_1 \sim A_N$, 서강대학교의 구역은 $B_1 \sim B_N$ 으로 구분되어 있다. A_i 구역에 a_i cm 만큼 비를 내리게 하고, B_j 구역에는 b_j cm 만큼 비를 내리게 할 것이다. 여기서 나오는 a_i 와 b_j 는 음이 아닌 실수다.

그러나 국렬이는 연세대학교 학생이고 인성이 매우 나쁘다. 때문에 비겁하게 연세대학교에 비를 더 많이 내리게 할 것이다. 당연히 서강대학교 측에서는 항의가 올 것이기 때문에 $a_i - b_j$ cm가 $c_{i,j}$ cm를 넘기지 않게 비를 내리게 할 것이다. 나쁜 국렬이는 최대한 연세대학교가 이익을 보길 원했기 때문에 $\sum_{i=1}^N a_i - \sum_{j=1}^N b_j$ cm가 최대가 되길 원했다. 이 최댓값을 구하여라.

입력

첫 번째 줄에는 $N(1 \leq N \leq 200)$ 이 주어진다.

두 번째 줄부터 $N+1$ 번째 줄까지, N 개의 양의 정수가 주어진다. $(i+1)$ 번째 줄의 j 번째 정수는 $c_{i,j}$ 를 의미한다. ($1 \leq c_{i,j} \leq 100$)

출력

$\sum_{i=1}^N a_i - \sum_{j=1}^N b_j$ 의 합의 최댓값을 소수 첫째 자리에서 반올림해 출력한다.

입출력 예시

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

$a_1 = 3, a_2 = 2, b_1 = 1, b_2 = 2$ 일 때 값이 최대가 된다.

표준 입력(stdin)	표준 출력(stdout)
3 2 3 2 3 4 4 2 3 2	8

$a_1 = 4, a_2 = 5, a_3 = 4, b_1 = 2, b_2 = 1, b_3 = 2$ 일 때 값이 최대가 된다.

이 페이지는 공백입니다

문제 E. 독특한 계산기

시간 제한 1 초
메모리 제한 1024 MB

당신은 수식을 독특한 방식으로 계산해야 한다. 수식을 계산하는 방식은 다음과 같다.

1. 수식에서 맨 앞의 연산자, 또는 맨 뒤의 연산자 먼저 계산한다. 단, 음수의 부호는 연산자로 취급하지 않는다.
2. 곱셈, 나눗셈을 덧셈, 뺄셈보다 더 먼저 계산한다.
3. 연산자의 우선순위가 같다면 해당 연산자를 계산했을 때 결과가 큰 것부터 계산한다.
4. 계산했을 때 결과 값 또한 같다면 앞에 것을 먼저 계산한다.

예를 들어서 수식이 $3 \times 2 + 5 - 5 + 7$ 으로 주어진다고 하면 다음과 같이 계산된다.

1. 3×2 와 $5 + 7$ 중에서 계산 우선순위가 더 높은 $\times$ 를 먼저 계산한다. 이후 계산식은 $6 + 5 - 5 + 7$ 이다.
2. 앞뒤의 연산자가 같으므로 $6 + 5$ 와 $5 + 7$ 을 비교했을 때, $5 + 7$ 이 더 크기 때문에 뒤에 있는 $+$ 를 먼저 계산한다. 이후 계산식은 $6 + 5 - 12$ 이다.
3. 뺄셈과 덧셈의 우선순위가 같으므로 $6 + 5$ 와 $5 - 12$ 를 비교했을 때, $6 + 5$ 가 더 크기 때문에 $+$ 를 먼저 계산한다. 이후 계산식은 $11 - 12$ 가 된다.
4. $11 - 12$ 를 계산하면 최종 결과 값은 -1 이 된다.

수식은 반드시 수와 연산자가 번갈아 가면서 나온다. 마지막에 연산자가 있는 경우와 맨 앞을 제외하고 음수가 들어오는 경우는 존재하지 않는다. 즉, $-1 - 1$ 같은 경우는 나올 수 있으나, $2 + -3$ 같은 경우는 존재하지 않는다고 가정해도 된다. 그리고 불필요한 0이 앞에 있을 수 있다. 즉, $001 + 0002$ 같은 수식이 나올 수 있다.

또한, 이 문제에서의 나눗셈은 C++에서 정수 간에 정의된 나눗셈으로 생각한다. 즉, 나누어지는 수가 양수면 나머지가 0 이상, 음수면 나머지가 0 이하로 처리가 되는 식으로 진행했을 때 나오는 몫을 계산하는 방식으로 이루어진다. 예를 들어, $3/2 = 1$, $(-3)/2 = -1$, $3/(-2) = -1$, $(-3)/(-2) = 1$ 로 계산된다.

이와 같은 계산 과정에 따라 주어진 식을 계산하시오.

입력

길이가 10^6 이하인 수식이 주어진다. 계산 과정 중의 모든 수는 -2^{63} 이상 2^{63} 미만이며, 0으로 나누는 경우는 없다. 숫자 앞에 불필요한 0이 있을 수 있다.

출력

주어진 식을 계산한 결과 값을 출력한다. 불필요한 0은 제거해야 한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
$3 \times 2 + 5 - 5 + 7$	-1
123456789	123456789
$3/2 \times 5 - 1 + 2 \times 90$	-176
$-1 - 1 - 1 - 1$	-2

처음에는 $-1 - 1 = -2$, $1 - 1 = 0$ 이기 때문에 $-1 - 1 - 0$ 이 된다.

두 번째는 $-1 - 1 = -2$, $1 - 0 = 1$ 이 되므로 $-1 - 1$ 이 된다.

마지막에는 $-1 - 1$ 을 계산해서 -2 가 된다.

이 페이지는 공백입니다

문제 F. 울타리

시간 제한 1 초
메모리 제한 1024 MB

준석이는 목장 N 채를 가지고 있다. 각 목장은 좌표평면 위의 점으로 나타낼 수 있다. 두 목장이 같은 위치에 있을 수 있다.

준석이는 모든 목장을 둘러싸는 직사각형 모양의 울타리를 치려고 한다. 직사각형의 변이 x 축이나 y 축에 꼭 평행할 필요는 없고, 변 위에 목장이 놓여 있어도 된다. 목장의 너비와 높이가 0이어도 된다.

모든 목장을 둘러싸는 울타리를 세웠을 때, 울타리의 최소 둘레를 구하여라.

입력

첫번째 줄에 목장의 수 $N(2 \leq N \leq 50,000)$ 이 주어진다.

다음 N 줄에는 한 줄에 하나씩 목장의 x 좌표와 y 좌표가 주어진다. 모든 좌표는 절댓값이 2×10^8 를 넘지 않는 정수이다.

출력

필요한 울타리의 최소 둘레를 출력한다. 절대/상대 오차는 10^{-7} 까지 허용한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
4 0 0 0 1 1 1 1 0	4

이 페이지는 공백입니다

문제 G. 마스크가 필요해

시간 제한 3 초
메모리 제한 1024 MB

코로나19가 발생하고 나서 마스크의 필요성이 증가하기 시작했다. 마스크가 없으면 생활을 하는데 어려움이 있기 때문에 마스크를 항상 구비하고 있어야 한다.

마스크의 수요가 증가함에 따라 일정했던 마스크의 가격이 상점마다 달라지게 되었다. 마스크의 가격이 제각각이기 때문에 A도시의 시민들이 전부 마스크를 갖기가 어려워지기 시작했다. A도시의 공무원인 당신은 이 사태를 해결하기 위해서 상점의 주인들에게 마스크의 가격을 일정하게 해달라고 했지만, 상점 주인들은 당연히 무시했다.

상점 주인들을 설득시키는 것은 어렵다고 생각해서 당신은 최대한 많은 시민들이 마스크를 가질 수 있게 하는 것으로 계획을 변경했다. 당신은 A도시의 각 시민이 마스크에 소비할 수 있는 금액의 범위와 A도시의 상점에서 팔고 있는 마스크의 가격 및 개수를 알아냈다. 이 정보를 바탕으로 최대한 많은 시민들이 마스크를 얻을 수 있게 하자.

입력

첫 번째 줄에는 A도시의 시민 수인 N 과 A도시의 상점 수인 M 이 주어진다. ($1 \leq N, M \leq 500,000$)

두 번째 줄부터 $N+1$ 번째 줄까지는 A도시의 각 시민이 마스크에 소비할 수 있는 돈의 범위인 L_i, R_i 가 주어진다. 즉 i 번째 A도시 시민이 살 수 있는 마스크의 가격은 L_i 이상 R_i 이하이다. ($1 \leq L_i \leq R_i \leq 10^{18}$)

$N+2$ 번째 줄부터 $N+M+1$ 번째 줄까지는 A도시의 각 상점이 판매하고 있는 마스크의 가격인 P_j 와 마스크의 개수인 X_j 가 주어진다. ($1 \leq P_j \leq 10^{18}, 1 \leq X_j \leq 1,000$)

모든 L_i, R_i, P_j, X_j 는 정수이다.

출력

가능한 한 많은 시민이 마스크를 샀을 때, 마스크를 산 시민의 수를 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
5 1 1 2 1 3 1 4 1 5 1 6 1 5	5
3 2 2 5 7 8 4 8 10 5 6 5	1
3 3 3 5 10 15 5 10 4 1 5 1 16 3	2

이 페이지는 공백입니다

문제 H. 전설

시간 제한 3 초
메모리 제한 1024 MB

Sogang ICPC Team에는 색상 이름과 닉네임의 순서로 이어서 팀명을 지으면 ICPC 리저널에서 수상할 수 있다는 전설이 있다. 색상 이름들과 닉네임들이 주어질 때, Q 개의 팀에 대해 다음 리저널에서 수상할 수 있을지 전설에 기반해 알려주는 프로그램을 작성하자.

입력

첫째 줄에는 색상의 종류 C 와 닉네임의 개수 N 이 주어진다. ($1 \leq C, N \leq 4,000$)

다음 C 개의 줄에는 색상 이름 C 개가 한 줄에 하나씩 주어진다. 색상 이름은 중복되지 않는다.

다음 N 개의 줄에는 Sogang ICPC Team 구성원들의 닉네임 N 개가 한 줄에 하나씩 주어진다. 닉네임도 중복되지 않는다.

다음 줄에는 팀의 개수 Q 가 주어진다. ($1 \leq Q \leq 20,000$)

다음 Q 개의 줄에는 팀명 Q 개가 한 줄에 하나씩 주어진다. 팀명은 중복되지 않는다.

모든 색상 이름의 길이와 닉네임의 길이는 1,000글자를 넘지 않는다. 모든 팀명은 2,000글자를 넘지 않는다. 모든 문자열은 알파벳 소문자로만 이루어져 있다.

출력

팀명이 입력된 순서대로, 전설에 따라 해당 팀이 다음 리저널에서 수상할 수 있다면 **Yes**, 아니라면 **No**를 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
4 3 red blue purple orange shift joker noon 5 redshift bluejoker purplenoon orangeshift whiteblue	Yes Yes Yes Yes No

이 페이지는 공백입니다

문제 I. 수학은 재밌어

시간 제한 1 초
메모리 제한 1024 MB

오일러는 수학을 정말 좋아해서 하루 종일 수학 공부만 하는 수학쟁이이다.

어느 날 오일러는 수학 공부를 하기 위해서 수학 책을 읽고 있던 중에 오일러 피 함수에 대해서 설명하는 부분을 보게 되었다. 오일러 피 함수는 다음과 같이 설명이 되어 있었다.

오일러 피 함수란 $\varphi(n)$ 으로 표기하며 1부터 n 까지의 양의 정수 중에서 n 과 서로소인 수의 개수를 나타내는 함수이다.

예를 들면 $\varphi(6)$ 은 1부터 6까지의 수 중 6과 서로소인 수의 개수를 말하는데 이는 1과 5로 두 개가 있으므로 $\varphi(6) = 2$ 이다.

오일러는 책의 내용을 곰곰이 읽던 중 어떤 문제가 떠올랐다. 문제의 내용은 다음과 같다.

어떤 양의 정수 n 이 있다고 할 때, $x\varphi(x) = n$ 을 만족하는 양의 정수 x 가 존재하는가?

고민에 빠진 오일러를 본 당신은 오일러의 궁금증을 해결해주기 위해서 직접 문제를 풀기로 결심했다. 그러므로 당신은 $x\varphi(x) = n$ 을 만족하는 x 를 구하는 프로그램을 작성하면 된다.

입력

첫 번째 줄에 n 이 입력으로 주어진다. ($1 \leq n \leq 10^9$)

출력

$x\varphi(x) = n$ 을 만족하는 양의 정수 x 가 존재하면 최소의 x 를, 존재하지 않으면 -1 을 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
2	2
3	-1
20	5

이 페이지는 공백입니다

문제 J. 상품권 준비

시간 제한 2 초
메모리 제한 1024 MB

2120년에도 “2120 신춘지역 대학생 프로그래밍 대회 동아리 연합 여름대회”가 성공적으로 개최된다. 올해 HI-ARC에서는 총 N 명의 학회원이 참가를 희망한다. HI-ARC의 운영진들은 서로 다른 정수로 수치화된 모든 학회원의 “실력” 정보를 가지고 있으며, 이를 바탕으로 다음과 같은 방법으로 팀을 구성한다.

- 정확히 a 개의 팀이 대회에 참가하며, 각 팀의 팀원은 M 명이다.
- 실력이 높은 상위 b 명은 이번 대회의 출제진이 된다. 따라서 남은 $N - b$ 명의 학회원이 이번 대회에 참가할 수 있는 후보다.
- 후보 중 $M \times a$ 명을 골라 최강의 팀 구성을 만든다. 한 팀의 “강력함”은 그 팀의 모든 팀원들의 실력의 곱으로 정의된다. 최강의 팀 구성은 모든 팀의 “강력함”의 합이 최대가 되는 구성이다.

HI-ARC에서는 대회 참가의 독려를 위해 모든 팀에게 소정의 상품권을 제공하는 이벤트를 준비하기로 하였다. 각 팀이 받는 상품권의 액수는, **팀원들의 이름을 Bitwise XOR한 값과 같다**. 알다시피 2120년 대한민국에서 모든 사람의 이름은 10^9 이하의 자연수이다.

문제는 한 팀의 팀원 수인 M 은 이미 결정되었지만, 아직 a 와 b 는 확정되지 않았다는 것이다. 따라서 HI-ARC 운영진들은 상품권을 총 얼마나 준비해야 할지 예산 계획을 세우는 데 어려움을 겪고 있다. 대회 준비로 바쁜 운영진들을 대신하여, 다양한 a 와 b 의 후보에 대해서 준비해야 하는 상품권 액수의 총합을 구해보자.

입력

첫 번째 줄에 대회에 참가하는 학회원의 수 N 과 한 팀의 팀원 수 M 가 주어진다. ($1 \leq M \leq N \leq 100,000$)

다음 N 개의 줄에 걸쳐 각 학회원의 정보를 나타내는 정수 x, y 가 주어진다. x 는 이 학회원의 실력, y 는 이 학회원의 이름이다. ($1 \leq x, y \leq 10^9$) 서로 다른 두 학회원의 x 가 같은 경우는 없다.

다음 줄에 쿼리의 개수 Q 가 주어진다. ($1 \leq Q \leq 500,000$)

다음 Q 개의 줄에 쿼리의 정보를 나타내는 정수 a_i 와 b_i 가 주어진다. ($1 \leq a_i \leq N, 0 \leq b_i < N$) 최강의 팀 구성을 만드는 방법이 하나인 쿼리들만 주어진다.

입력의 양이 많은 편이므로 빠른 입력 함수의 사용을 권장한다.

출력

i ($1 \leq i \leq Q$)번째 줄에 $a = a_i$ 이고 $b = b_i$ 일 때 준비해야 하는 상품권 액수의 총합을 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
5 2 10 2 20 7 30 6 40 8 50 4 1 2 1	19

“4”가 출제진이 되어야 하며, [“2”, “7”], [“6”, “8”]로 팀을 구성하는 것이 ($10 \times 20 + 30 \times 40$)으로 최강의 팀 구성이다. 따라서 총 $(2 \oplus 7) + (6 \oplus 8) = 19$ 가 준비해야 할 총액이다.

노트

Bitwise XOR은 두 값을 이진수로 표현하고 자리 단위로 적용되는 이진 연산자로, 두 피연산자의 각 자릿수를 비교하며 같으면 1, 다르면 0을 계산한다. C/C++, Java, Python에서 두 정수 x 와 y 의 Bitwise XOR 결과는 $(x \oplus y)$ 로 계산할 수 있다.

문제 K. 물건 가져가기

시간 제한 1 초
메모리 제한 1024 MB

국렬이는 신촌 어딘가에 있는 물건을 얻을 수 있는 사격장에 갔다. 해당 사격장에 물품이 N 개가 있고 각 물품 별로 번호가 1부터 N 까지 붙어 있다. 물품들을 총으로 맞추면 해당 물품을 가져갈 수 있다.

그러나 물건을 가져가는 것에 대해서 제한이 있는데, 각 물품마다 사전에 얻어야 하는 물품들이 있는데 해당 물품들을 전부 얻어야 그 물품을 가져갈 수 있다. 그렇지 않으면 해당 물품을 맞춰도 가져갈 수 없다. 예를 들면, A라는 물건은 B와 C 두 개를 전부 얻어야만 가져갈 수 있고, C는 D를 먼저 얻어야 가져갈 수 있다면, A를 가져가기 위해서는 B, C, D를 전부 얻고 나서 가져갈 수 있다.

다만 쏘는 시점은 상관없다. 예를 들어, A는 B를 얻어야 가져갈 수 있고, B는 A를 얻어야 가져갈 수 있다면, B를 먼저 얻고 A를 가져가든, A를 먼저 얻고 B를 가져가든 상관없다. 즉, 이런 경우 둘 다 가져가지 않거나, 둘 다 가져가야 한다는 뜻이다.

각 물품들을 얻었을 때, 국렬이의 기분의 변화 정도가 있다. 얻었을 때 역겨움과 혐오감이 몰려와서 기분이 낮아질 수 있고, 반대로 환희감이 차올라서 기분이 높아질 수 있다.

총알은 각 물품 개수만큼 있고, 국렬이는 사격의 마스터라 원하는 물건은 무조건 맞출 수 있다. 이때, 국렬이의 기분의 최대치를 얻을 수 있는 물건들을 구하여라.

입력

첫 줄에는 물품의 개수와 물품 간의 관계의 수를 나타내는 N 과 M 이 주어진다. ($1 \leq N \leq 200, 1 \leq M \leq 400$)

두 번째 줄부터 $N+1$ 번째 줄까지는 각 물품을 얻었을 때 기분의 변화치 t_i 를 나타내는 정수가 주어진다. ($100,000 \leq t_i \leq 100,000$)

$N+2$ 번째 줄부터 $N+M+1$ 번째 줄까지 물품들 간의 관계에 대한 서로 다른 정수 2개 s_i, e_i 가 주어진다. s_i 의 물품을 얻기 위해서는 e_i 를 얻어야 한다는 의미다. ($1 \leq s_i, e_i \leq N$)

출력

얻어야 하는 물품의 개수를 첫 줄에 출력하고, 다음 줄에 각 물품들의 번호를 공백으로 구분해 출력하라.

답이 여러 개인 경우에는 그 중 아무거나 출력하라.

입출력 예시

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

이 페이지는 공백입니다

문제 L. 카드 셔플

시간 제한 10 초
메모리 제한 1024 MB

위에서부터 1, 2, ..., N 이 적혀 있는 카드 한 벌이 있다. 이 때, 다음 쿼리를 수행한다.

- **1 x y**: 위에서부터 x 번째에 있는 카드부터 y 번째에 있는 카드까지를 제일 위로 올린다.
- **2 x y**: 위에서부터 x 번째에 있는 카드부터 y 번째에 있는 카드까지를 제일 아래로 내린다.
- **3 x y**: 위에서부터 x 번째에 있는 카드부터 y 번째에 있는 카드까지 **리플 셔플**을 한다. $y - x + 1$ 은 1,000 이하이다.

리플 셔플은 N 장의 카드가 있을 때 카드를 위에서부터 $(N+1)/2$ 장의 카드 덩어리와 나머지 덩어리로 나눈 후 교대로 카드를 두는 방식으로 카드를 섞는 것이다. 섞기 전에 제일 위에 있었던 카드가 섞은 후에도 제일 위로 오도록 한다.

입력

첫 줄에 카드 수 N ($1 \leq N \leq 10^6$)과 쿼리의 수 Q ($1 \leq Q \leq 2 \times 10^5$)가 주어진다. 다음 Q 개의 줄에 쿼리가 주어진다.

출력

카드를 다 섞고 난 뒤 카드에 어떤 숫자가 적혀 있는지 위에서부터 순서대로 출력한다.

입출력 예시

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

노트

카드 6장이 있을 때 쿼리 **3 1 6**을 처리하면

$$1\ 2\ 3\ 4\ 5\ 6 \Rightarrow 1\ 4\ 2\ 5\ 3\ 6$$

이 되고, 카드 7장이 있을 때 **3 1 7**이 입력으로 들어오면

$$1\ 2\ 3\ 4\ 5\ 6\ 7 \Rightarrow 1\ 5\ 2\ 6\ 3\ 7\ 4$$

가 된다.

이 페이지는 공백입니다