

## 문제 1. 시험

입력 파일: standard input  
출력 파일: standard output  
시간 제한: 3 seconds  
메모리 제한: 1024 megabytes

$N$ 명에 학생이 수학 부문과 정보 부문이 있는 시험을 쳤다.  $i$ 번째 ( $1 \leq i \leq N$ ) 학생은 수학에서는  $S_i$ 점을, 정보에서는  $T_i$ 점을 받았다. T교수와 I교수는 각 학생이 시험을 통과할지 말지를, 점수를 기반으로 정하려고 한다.

- T 교수는 두 과목을 모두 중요하게 본다. 수학에서  $A$ 점, 정보에서  $B$ 점을 받아야만 통과한 것으로 생각한다.
- I 교수는 총점만 중요하게 본다. 수학과 정보를 합쳐서  $C$ 점 받아야 통과한 것으로 생각한다.
- 두 교수의 기준을 모두 통과한 학생만 시험을 통과할 수 있다.

당신은 기준인  $A, B, C$ 를 모른다. 하지만,  $Q$ 개의 정수 ( $X_j, Y_j, Z_j$ ) ( $1 \leq j \leq Q$ ) 가 주어져서 몇 명의 학생들이  $A = X_j, B = Y_j, C = Z_j$ 일 때 시험을 통과하는지 알고 싶다.

학생들의 수, 점수 정보와 점수 기준이 주어졌을 때, 이 조건 하에서 시험을 통과하는 학생의 수를 구하여라.

### 입력 형식

표준 입력에서 다음과 같은 형식으로 주어진다. 모든 값은 정수이다.

$N \ Q$

$S_1 \ T_1$

$\vdots$

$S_N \ T_N$

$X_1 \ Y_1 \ Z_1$

$\vdots$

$X_Q \ Y_Q \ Z_Q$

### 출력 형식

표준 출력으로  $Q$ 개의 줄을 출력하여라.  $j$ 번째 ( $1 \leq j \leq Q$ )줄은 몇 명의 학생들이  $A = X_j, B = Y_j, C = Z_j$ 일 때 시험을 통과하는 학생 수이다.

### 제한

- $1 \leq N \leq 100\ 000$ .
- $1 \leq Q \leq 100\ 000$
- $0 \leq S_i \leq 1\ 000\ 000\ 000$  ( $1 \leq i \leq N$ ).
- $0 \leq T_i \leq 1\ 000\ 000\ 000$  ( $1 \leq i \leq N$ ).
- $0 \leq X_j \leq 1\ 000\ 000\ 000$  ( $1 \leq j \leq Q$ ).
- $0 \leq Y_j \leq 1\ 000\ 000\ 000$  ( $1 \leq j \leq Q$ ).
- $0 \leq Z_j \leq 2\ 000\ 000\ 000$  ( $1 \leq j \leq Q$ ).

### 서브태스크 1 (2 점)

- $N \leq 3\,000$
- $Q \leq 3\,000$

### 서브태스크 2 (20 점)

- $S_i \leq 100\,000$  ( $1 \leq i \leq N$ ).
- $T_i \leq 100\,000$  ( $1 \leq i \leq N$ ).
- $X_j \leq 100\,000$  ( $1 \leq j \leq Q$ ).
- $Y_j \leq 100\,000$  ( $1 \leq j \leq Q$ ).
- $Z_j = 0$  ( $1 \leq j \leq Q$ ).

### 서브태스크 3 (21 점)

- $S_i \leq 100\,000$  ( $1 \leq i \leq N$ ).
- $T_i \leq 100\,000$  ( $1 \leq i \leq N$ ).
- $X_j \leq 100\,000$  ( $1 \leq j \leq Q$ ).
- $Y_j \leq 100\,000$  ( $1 \leq j \leq Q$ ).
- $Z_j \leq 200\,000$  ( $1 \leq j \leq Q$ ).

### 서브태스크 4 (57 점)

추가 제한조건이 없다.

## 예제

standard input	standard output
5 4	2
35 100	4
70 70	1
45 15	1
80 40	
20 95	
20 50 120	
10 10 100	
60 60 80	
0 100 100	
10 10	1
41304 98327	3
91921 28251	5
85635 59191	8
30361 72671	8
28949 96958	3
99041 37826	3
10245 2726	3
19387 20282	5
60366 87723	6
95388 49726	
52302 69501 66009	
43754 45346 3158	
25224 58881 18727	
7298 24412 63782	
24107 10583 61508	
65025 29140 7278	
36104 56758 2775	
23126 67608 122051	
56910 17272 62933	
39675 15874 117117	

## 참고 사항

첫째 예제에서

- $A = 20$ ,  $B = 50$ ,  $C = 120$ 일 때, 첫 번째와 두 번째 학생만 수학 부문에서 최소 20점, 정보 시험에서 최소 50점, 그리고 총점 120점을 넘길 수 있다. 그래서 시험을 통과하는 학생들의 수는 2이다.
- $A = 10$ ,  $B = 10$ ,  $C = 100$ 일 때, 첫 번째, 두 번째, 네 번째 그리고 다섯 번째 학생만 수학 부문에서 최소 10점, 정보 시험에서 최소 10점, 그리고 총점 100점을 넘길 수 있다. 그래서 시험을 통과하는 학생들의 수는 4이다.
- $A = 60$ ,  $B = 60$ ,  $C = 80$ 일 때, 두 번째 학생만 수학 부문에서 최소 60점, 정보 시험에서 최소 60점, 그리고 총점 80점을 넘길 수 있다. 그래서 시험을 통과하는 학생들의 수는 1이다.
- $A = 0$ ,  $B = 100$ ,  $C = 100$ 일 때, 첫 번째 학생만 수학 부문에서 최소 0점, 정보 시험에서 최소 100점, 그리고 총점 100점을 넘길 수 있다. 그래서 시험을 통과하는 학생들의 수는 1이다.

## 문제 2. 비버의 모임

입력 파일: standard input  
출력 파일: standard output  
시간 제한: 2 seconds  
메모리 제한: 256 megabytes

0부터  $N-1$ 까지의 번호가 붙은 비버가 살고 있는  $N$ 개의 섬이 있다. 각 섬은  $N-1$ 개의 양방향으로 연결되는 다리로 연결되어 있다. 몇 개의 다리를 사용하면 어떠한 두 섬을 오가는것도 가능하다. **각 섬마다, 섬에 직접 연결된 다리는 최대 18개 이다.** 각 섬에는 비버가 살고 있다.

가끔, 몇몇 비버는 특정 섬에서 모임을 한다. 세 비버가 만날 때, 그들은 다음과 같은 조건을 만족하는 섬에서 만난다:

- 모임을 할 때, 3마리의 비버가 자신이 살고 있는 섬부터 이동할 때 사용하는 다리의 수의 합이 최소가 되는 섬 (이런 섬은 유일하게 존재한다.)

이 섬은 3마리의 비버중 하나가 살고 있는 섬일 수 있음에 주의하여라.

당신은  $N$ 개의 섬이 어떤 방식으로 다리로 연결되어 있는지가 궁금해 졌다. 당신은 직접 이 섬을 확인할 수 없다. 그렇기 때문에, 당신은 비버에게 명령을 내리기로 했다. 명령은 다음과 같다.

- 세 개의 섬  $u, v, w$  ( $0 \leq u \leq N-1, 0 \leq v \leq N-1, 0 \leq w \leq N-1, u \neq v, u \neq w, v \neq w$ )을 지정하고  $u, v, w$ 에 사는 비버끼리 모임을 갖게 한다.
- 비버가 모임을 갖는 섬을 알 수 있다.

당신은 섬이 어떤 방식으로 다리로 연결되어 있는지를 적은 수의 명령으로 알고 싶다. 섬의 수가 주어졌고, 비버와 통신하는 방법이 주어졌을 때, 섬이 연결되어 있는 방식을 알아내어라.

## 구현 명세

당신은 파일 하나를 제출해야 한다.

이 파일의 이름은 `meetings.cpp`이다. 파일은 다음 함수를 구현해야 한다. 또한, `meetings.h`를 `include`해야 한다.

- `void Solve(int N)`
  - 이 함수는 각 테스트 케이스마다 정확히 한 번 불린다.
  - 인자  $N$ 은 섬의 수  $N$ 을 나타낸다.당신의 프로그램은 다음 함수를 호출 할 수 있다.
  - `int Query(int u, int v, int w)`
    - 이 함수는, 주어진 세 개의 섬에 대해서 세 비버들이 만나는 섬의 번호를 반환한다.
    - \* 당신은 섬의 번호  $u, v, w$ 를 인자  $u, v, w$ 를 사용해서 나타내어야 한다. 이 번호는  $0 \leq u \leq N-1, 0 \leq v \leq N-1, 0 \leq w \leq N-1, u \neq v, u \neq w, v \neq w$ 을 모두 만족해야 한다. 아닌 경우에는, **오답 [1]**이 된다.
    - \* 당신은 이 함수를 100 000번 이상 호출해서는 안된다. 호출 한 경우에는 **오답 [2]**이 된다.
  - `void Bridge(int u, int v)`
    - 이 함수는, 섬이 다리로 연결되어 있는지에 대한 정보를 줄 수 있다.
    - \* 인자  $u$ 와  $v$ 는 섬  $u$ 와 섬  $v$ 가 다리로 서로 직접 연결되어있다는 것을 나타낸다.
    - \*  $0 \leq u < v \leq N-1$ 이 아닌 경우, **오답 [3]**이 된다.
    - \* 섬  $u$ 와  $v$ 가 직접 다리로 연결되어있지 않은 경우 **오답 [4]**이 된다.

- \* 함수가 같은 인자  $u, v$ 를 여러번 호출 한 경우 **오답 [5]**이 된다.
- \*  $N-1$ 개의 다리가 있으므로, 함수는 정확히  $N-1$ 번 호출 되어야 한다. 만약 함수 `Solve`가 끝날 때 이 함수의 호출 횟수가  $N-1$ 이 아니면, **오답 [6]**이 된다.

## 참고 사항

- 당신의 프로그램은 내부에서 사용할 목적으로 함수나 전역변수를 사용할 수 있다.
- 당신의 프로그램은 표준 입출력을 사용해서는 안된다. 당신의 프로그램은 어떠한 방법으로든 다른 파일에 접근해서는 안된다. 단, 당신의 프로그램은 디버그 목적으로 표준 에러출력에 출력할 수 있다.

당신은 대회 홈페이지의 아카이브에서 프로그램을 테스트 하기 위한 목적의 샘플 그레이더를 받을 수 있다. 아카이브는 당신의 프로그램의 예제 소스 또한 첨부되어 있다. 샘플 그레이더는 파일 `grader.cpp`이다. 당신의 프로그램을 테스트 하기 위해서, `grader.cpp`, `meetings.cpp`, `meetings.h`를 같은 디렉토리 안에 놓고, 컴파일 하기 위해 다음 커맨드를 실행하여라.

- `g++ -std=gnu++14 -O2 -o grader grader.cpp meetings.cpp`

컴파일이 성공적이면, 파일 `grader`가 생성된다.

실제 그레이더와 샘플 그레이더는 다름에 주의하여라. 샘플 그레이더는 하나의 프로세스에서 실행 되며, 입력을 표준 입력으로 부터 받고, 출력을 표준 출력에 출력한다.

## 입력 형식

샘플 그레이더는 표준 입력에서 다음과 같은 형식으로 입력받는다.

$N$   
 $A_0 B_0$   
 $\vdots$   
 $A_{N-2} B_{N-2}$

$A_i$ 와  $B_i$  ( $0 \leq i \leq N-2$ )는 섬  $A_i$ 와  $B_i$ 가 다리로 직접 연결되어있다는 것을 의미한다.

## 출력 형식

프로그램이 정상적으로 종료되었다면, 샘플 그레이더는 다음과 같은 정보를 표준 출력에 출력한다. (따옴표는 출력하지 않는다.)

- 정답으로 판단 된 경우, Query함수의 호출 횟수를 “Accepted: 100”과 같은 형식으로 출력한다.
- 오답으로 판단 된 경우, 오답의 종류를 “Wrong Answer [1]”과 같은 형식으로 출력한다.

프로그램이 다양한 오답의 종류에 속해 있을 경우, 샘플 그레이더는 그 중 하나만 출력 할 것이다.

## 제한

샘플 그레이더의  $A_i$ 와  $B_i$ 의 정의에 따라서

- $3 \leq M \leq 2\,000$ .
- $0 \leq A_i < B_i \leq N-1$ . ( $0 \leq i \leq N-2$ )
- 몇 개의 다리를 사용하면 어떠한 두 섬을 오가는것도 가능하다.
- 각 섬 마다, 섬에 직접 연결된 다리는 최대 18개 이다.

## 서브태스크 1 (7 점)

- $N \leq 7$

## 서브태스크 2 (10 점)

- $N \leq 50$

## 서브태스크 3 (12 점)

- $N \leq 300$

## 서브태스크 4 (71 점)

추가 제한조건이 없다.

- 서브태스크 1, 2, 3에 대해서, 서브태스크 안에 있는 모든 테스트 케이스를 맞춘 경우 점수를 준다.
- 서브태스크 4에 대해서,  $X$ 를 Query함수의 최대 호출횟수라고 하자.
  - $40\,000 < X \leq 100\,000$ 이면 49점을 받는다.
  - $X \leq 40\,000$ 이면 71점을 받는다.

## 예제

이 함수는 그레이더의 예제 입력과 해당하는 함수 호출을 보여준다.

예제 입력	예제 함수 호출		
	호출	호출	반환값
5 0 1 0 2 1 3 1 4	Solve(5)		
		Query(0, 1, 2)	0
		Query(0, 3, 4)	1
		Bridge(1, 3)	(없음)
		Bridge(0, 2)	(없음)
		Bridge(1, 4)	(없음)
		Bridge(0, 1)	(없음)

## 문제 3. 난

입력 파일:	standard input
출력 파일:	standard output
시간 제한:	4 seconds
메모리 제한:	256 megabytes

JOI 카레 매점은 매우 긴 난(인도의 납작한 빵)을 판매하는 것으로 유명하다. 난에는  $L$ 개의 맛이 있으며, 1번부터  $L$ 번까지 번호가 붙어 있다. 난 중에서 “JOI 스페셜 난”이 제일 인기가 있다. 길이가  $L\text{cm}$  이고, 왼쪽에서  $j - 1\text{cm}$ 부터  $j\text{cm}$ 까지 부분에는  $j$ 번 ( $1 \leq j \leq L$  맛으로 되어 있다).

$N$ 명의 사람이 JOI 카레 매점에 왔다. 그들의 취향은 다른 사람과 다르다. 구체적으로,  $i$  번째 ( $1 \leq i \leq N$ ) 사람이  $j$ 번 ( $1 \leq j \leq L$ ) 맛의 난을 먹었을 경우에는,  $1\text{cm}$ 당  $V_{i,j}$ 의 행복도를 얻을 것이다. 그들은 하나의 JOI 스페셜 난을 주문 했다. 그들은 난을 다음과 같은 방법으로 나누어 가질 것이다.

1.  $0 < X_1 < X_2 < \dots < X_{N-1} < L$ 을 만족하는  $N - 1$ 개의 분수  $X_1, \dots, X_{N-1}$ 를 고른다.
2.  $N$ 개의 정수  $P_1, \dots, P_N$ 을 고른다. 이는  $1, \dots, N$ 의 순열이어야 한다.
3. 각  $k$  ( $1 \leq k \leq N - 1$ )에 대해서, 난을  $X_k$ 지점에서 자른다. 난은  $N$ 개의 조각으로 나누어 질 것이다.
4. 각  $k$  ( $1 \leq k \leq N$ )에 대해서,  $P_k$ 번째 사람에게  $X_{k-1}$ 과  $X_k$  사이의 조각을 준다. 우리는  $X_0$ 을  $0$ ,  $X_N$ 을  $L$ 이라고 생각 할 것이다.

우리는 난을 공평하게 나누고 싶다. 우리는 각 사람이 혼자 JOI 스페셜 난을 모두 먹었을 때 얻는 행복도의  $1/N$ 이상을 얻었을 경우, 분배 방식이 **공평하다** 라고 할것이다.

$N$ 명의 사람의 선호가 주어졌을 때, 난을 공평하게 나누는 방법이 있는가를 출력하여라. 있는 경우, 난을 공평하게 나누는 방법에 대해 출력하여라.

### 입력 형식

표준 입력에서 다음과 같은 형식으로 주어진다. 모든 수는 정수이다.

$N\ L$

$V_{1,1}\ V_{1,2}\ \dots\ V_{1,L}$

$\vdots$

$V_{N,1}\ V_{N,2}\ \dots\ V_{N,L}$

### 출력 형식

난을 공평하게 나누는 방법이 없다면,  $-1$ 을 첫째 줄에 출력하여라. 공평하게 나눌 수 있다면, 나누는 방법을 나타내는  $N - 1$ 개의 분수  $X_1, \dots, X_N$ 과  $N$ 개의 정수  $P_1, \dots, P_N$ 을 다음 형식으로 출력하여라.

$A_1\ B_1$

$A_2\ B_2$

$\vdots$

$A_{N-1}\ B_{N-1}$

$P_1\ P_2\ \dots\ P_N$

$A_i, B_i$ 는  $X_i = \frac{A_i}{B_i}$  ( $1 \leq i \leq N$ )를 만족하는 정수 쌍이다. 이 정수는 출력 제한을 따라야 한다.

### 제한

#### 입력 제한

- $1 \leq N \leq 2000$ .
- $0 \leq L \leq 2000$ .
- $1 \leq V_{i,j} \leq 100\,000$ . ( $1 \leq i \leq N$ ,  $1 \leq j \leq L$ )

### 출력 제한

난을 공평한 방식으로 나누는 방법이 존재한다면, 출력은 다음 제한을 따라야 한다.

- $1 \leq B_i \leq 1\,000\,000\,000$ . ( $1 \leq i \leq N$ )
- $0 \leq \frac{A_1}{B_1} < \frac{A_2}{B_2} < \dots < \frac{A_{N-1}}{B_{N-1}} < L$ .
- $P_1, \dots, P_N$ 은  $1, \dots, N$ 의 순열이다.
- 분배에서,  $i$ 번째 사람이 가지는 행복도의 양은  $\frac{V_{i,1} + V_{i,2} + \dots + V_{i,L}}{N}$  이상 이어야 한다.

$A_i$ 와  $B_i$ 는 서로소일 필요는 없다. 아래 제한 하에서, 공평한 분배가 존재 할 경우  $1 \leq B_i \leq 1\,000\,000\,000$ 을 만족하는 출력이 존재함을 증명할 수 있다.

### 서브태스크 1 (5 점)

- $N = 2$

### 서브태스크 2 (24 점)

- $N \leq 6$
- $V_{i,j} \leq 10$  ( $1 \leq i \leq N$ ,  $1 \leq j \leq L$ )

### 서브태스크 3 (71 점)

추가 제한조건이 없다.

### 예제

standard input	standard output
2 5 2 7 1 8 2 3 1 4 1 5	14 5 2 1

이 예제에서, 모든 난을 먹었을 때, 첫째 사람은  $2 + 7 + 1 + 8 + 2 = 20$ 의 행복도를 가지고 둘째 사람은  $3 + 1 + 4 + 1 + 5 = 14$ 의 행복도를 가진다. 즉, 첫째 사람이  $\frac{20}{2} = 10$  이상의 행복도를 가지고 둘째 사람이  $\frac{14}{2} = 7$  이상의 행복도를 가지면, 분배는 공평하다.

난을  $\frac{14}{5}$ 에서 나누면, 첫째 사람은  $1 \times \frac{1}{5} + 8 + 2 = \frac{51}{5}$ 의 행복도를 얻고, 둘째 사람은  $3 + 1 + 4 \times \frac{4}{5} = \frac{36}{5}$ 의 행복도를 얻는다. 그러므로, 이것은 공평한 분배이다.

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



이 예제에서는 맛이 하나 뿐이다. 난을 크기가 같은 7개의 부분으로 자르면,  $P_1, \dots, P_N$ 과 관계 없이 분배가 공정하다.

standard input	standard output
5 3	15 28
2 3 1	35 28
1 1 1	50 28
2 2 1	70 28
1 2 2	3 1 5 2 4
1 2 1	

$A_i$ 와  $B_i$ 가 서로소 일 필요는 없다. ( $1 \leq i \leq N$ )