#### 짝수번 겹치는 구간 구하기

수직선 상에 n개의 닫힌 구간(양쪽 끝점이 모두 포함된 구간)이 주어졌을 때, 수직선 상에서 짝수번 겹치는 구간을 모두 출력하는 프로그램을 작성하십시오. 구간의 정확한 정의와, 짝수번 겹치는 것의 정확한 의미는 아래에 추가 설명하겠습니다. 구간의 표기와, 해당 표기에 대한 의미는 다음과 같이 정의합니다.

예를 들어 [1, 3], [5, 6], [2, 4]라는 구간이 주어졌다면, 수직선 상에서 다음과 같이 나타낼 수 있습니다.

여러 구간이 한 점 이상을 공유하면 이 공유하는 구간을 ‘겹친다’고 표현합니다. 이 때, 수직선 상에서 겹치는 ‘입력 구간’이 짝수 개인 구간을 모두 구하면 됩니다. 위 예에서, 1.5라는 점은 [1, 3]이라는 1개 입력 구간만 겹치고 있기에 짝수번 겹치는 구간에 포함이 안 되지만, 2.5라는 점은 [1, 3], [2, 4]의 2개 입력 구간과 겹치므로 짝수번 겹치는 구간에 포함이 됩니다. 점 2부터 점 3까지는 모두 [1, 3], [2, 4]의 두 개 입력 구간과 겹치므로 [2, 3]이라는 구간이 바로 짝수번 겹치는 구간이 됩니다. 마찬가지로 점 1은 입력 구간 [1, 3]과 1번 겹치지만 점 1보다 조금이라도 작은 수는 주어진 입력 구간과 0번 겹치므로 짝수번 겹치는 구간에 포함이 됩니다. 이런 과정으로 짝수번 겹치는 구간을 구하면 (- 1), [2, 3], (4, 5), (6, +)가 됩니다. 숫자 n과, n개의 닫힌 구간을 나타내는 2n개의 자연수가(이 수는 1,000,000,000이하입니다) 표준 입력으로 주어졌을 때, 수직선 상에서 짝수번 겹치는 구간을 출력하는 프로그램을 작성하십시오. 결과의 출력 순서는 구간의 왼쪽 값의 오름차순 순으로 출력해야 하며, 출력하는 구간의 개수는 최소화해야 합니다. 예를 들어, 문제에서 알아낸 출력 구간이 (3, 5), [5, 7]과 같은 구간이 있다면 (3, 7)의 하나의 구간으로 합쳐서 출력해야 합니다. n개의 닫힌 구간 입력값은 모두 왼쪽 값이 오른쪽 값보다 작거나 같다고 가정해도 좋습니다. 입력된 n개 구간의 양 끝점이 모두 다르다는 가정은 해서는 안 됩니다. 출력 시에는 구간의 양 끝점의 포함 유무를 명확하게 구분하여 출력해야 합니다.

입/출력 예시

⋇ 입출력 형식을 잘 지켜주세요.

␣

 : 공백

↵

 : 줄바꿈

보기 입력 1

3↵  
1 ␣3↵  
5 ␣6↵  
2 ␣4

3  
1 3  
5 6  
2 4

출력 1

(-, ␣1)↵  
[2, ␣3]↵  
(4, ␣5)↵  
(6, ␣+)

(-, 1)  
[2, 3]  
(4, 5)  
(6, +)

//Please don't change class name 'Main'

import java.util.Scanner;

import java.util.\*;

class Main {

public static void main(String[] args) {

try(Scanner s = new Scanner(System.in))

{

int n = s.nextInt();

int[][] point = new int[2][n];

int numberOne = 0;

int numberTwo = 0;

for(int i = 0; i < n; i++)

{

int a = s.nextInt();

int b = s.nextInt();

//여기부터 작성해 주세요

point[0][i] = a;

point[1][i] = b;

}

int low = Integer.MAX\_VALUE;

int high = Integer.MIN\_VALUE;

for(int i = 0; i < n; i++){

if(point[0][i] < low){

low = point[0][i];

}

if(point[1][i] > high){

high = point[1][i];

}

}

int[] cnt = new int[high-low+1];

for(int i = 0 ; i < n ; i++){

for(int j = point[0][i] ; j <= point[1][i] ; j++){

cnt[j-1]++;

}

}

System.out.println("(-, " + low + ")");

for(int i = 1 ; i < cnt.length-1 ; i++){

if(cnt[i] % 2 == 0 && numberTwo == 0){

System.out.print("[" + (i+1) + ", ");

++numberTwo;

continue;

}

if(cnt[i] % 2 == 0 && numberTwo == 1){

System.out.println((i+1) + "]");

numberTwo--;

continue;

}

if(cnt[i] % 1 == 0 && numberOne == 0){

System.out.print("(" + (i+1) + ", ");

numberOne++;

continue;

}

if(cnt[i] % 1 == 0 && numberOne == 1){

System.out.println((i+1) + ")");

numberOne--;

continue;

}

}

System.out.println("(" + high + ", +)");

}

}

}