**2. Overlapping rectangles**

Given two rectangles, find if the given two rectangles overlap or not. A rectangle is denoted by providing the x and y coordinates of two points: the left top corner and the right bottom corner of the rectangle. Two rectangles sharing a side are considered overlapping. (L1 and R1 are the extreme points of the first rectangle and L2 and R2 are the extreme points of the second rectangle).

**Note:**It may be assumed that the rectangles are parallel to the coordinate axis.

// Initial Template for Java

import java.io.\*;

import java.util.\*;

class GFG {

public static void main(String args[]) throws IOException {

BufferedReader read =

new BufferedReader(new InputStreamReader(System.in));

int t = Integer.parseInt(read.readLine());

while (t-- > 0) {

String S[] = read.readLine().split(" ");

int p[] = new int[2];

int q[] = new int[2];

int r[] = new int[2];

int s[] = new int[2];

p[0] = Integer.parseInt(S[0]);

p[1] = Integer.parseInt(S[1]);

q[0] = Integer.parseInt(S[2]);

q[1] = Integer.parseInt(S[3]);

r[0] = Integer.parseInt(S[4]);

r[1] = Integer.parseInt(S[5]);

s[0] = Integer.parseInt(S[6]);

s[1] = Integer.parseInt(S[7]);

Solution ob = new Solution();

int ans = ob.doOverlap(p, q, r, s);

System.out.println(ans);

}

}

}// } Driver Code Ends

// User function Template for Java

**class Solution {**

**int doOverlap(int L1[], int R1[], int L2[], int R2[]) {**

**// code here**

**if(L1[0]>R2[0] || L2[0]>R1[0])**

**return 0;**

**if(L1[1]<R2[1] || L2[1]<R1[1])**

**return 0;**

**return 1;**

**}**

**};**