
Contest Rescheduling

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 512 megabytes

There are many programming competitions nowadays. For contestants' convenience competition organizers try to do their best to avoid time intersections for the competitions, but sometimes this happens.

Once two competitions were scheduled for the same day. The first competition was scheduled to start at s_1 with the duration of d_1 (ending at $s_1 + d_1$), and the second one at s_2 with the duration of d_2 (ending at $s_2 + d_2$). The duration of the competitions can't be changed, but the start time can. But there are restrictions for each competition: the first one shouldn't start before l_1 , and end after r_1 , the second one shouldn't start before l_2 , and end after r_2 .

Your task is to write a program that can help organizers to reschedule the competitions, so that they don't intersect by their time, and given restrictions are still satisfied. If there are several new schedules, you are required to find one that the total change is minimized. Formally speaking, if c_1 and c_2 are the start times in the new schedule, then $(|s_1 - c_1| + |s_2 - c_2|)$ has to be minimized.

Input

Input data consists of several testcases.

First line contains an integer n — the number of testcases to solve ($1 \leq n \leq 50\,000$).

The description of n tests follow, each of them consists of two lines.

The first of these lines contains four integers l_1, r_1, l_2 , and r_2 — the competitions' schedule restrictions ($0 \leq l_i < r_i \leq 10^9$).

The second one contains four integers s_1, d_1, s_2 , and d_2 — the starting time and duration of each competition ($l_i \leq s_i; s_i + d_i \leq r_i; d_i \geq 1$).

Output

Print n lines: the i -th line containing an answer for i -th testcase.

Print two integers c_1 and c_2 for each testcase: the starting time of the first and the second competition in a new schedule, respectively. If there are several solutions that minimize the total change, print any. If there is no way to make a new schedule, print two -1 .

Scoring

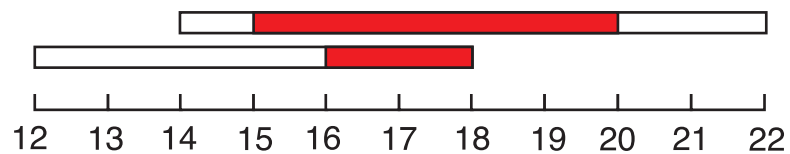
Subtask	Score	n	s_i, d_i	l_i, r_i
1	19	$n \leq 1000$	$s_i, d_i \leq 20$	$l_i = 0; r_i = 50$
2	20	$n \leq 1000$	$s_i, d_i \leq 50$	$l_i, r_i \leq 50$
3	21	$n \leq 1000$	$s_i, d_i \leq 10\,000$	$l_i, r_i \leq 10\,000$
4	15	$n \leq 1000$	$s_i, d_i \leq 10^9$	$l_i, r_i \leq 10^9$
5	25	$n \leq 50\,000$	$s_i, d_i \leq 10^9$	$l_i, r_i \leq 10^9$

Example

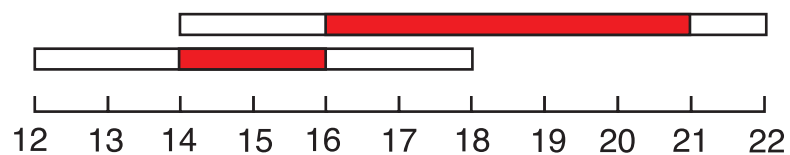
standard input	standard output
3	16 14
14 22 12 18	-1 -1
15 5 16 2	12 16
12 22 14 20	
14 5 15 4	
12 14 16 18	
12 2 16 2	

Explanation

Picture for the first testcase. Initial schedule:



New schedule:



The total change is $|15 - 16| + |16 - 14| = 3$. This is the minimum possible change.