Begin: 2021-10-11 12:30 CST

NCPC Simulation Day3

End: 2021-10-11 17:30 CST

Elapsed: 05:02:32 Running **Remaining:** -1:57:27

Time limit

2000 ms

Memory limit

262144 kB

C - Segment Intersections

You are given two lists of segments $[al_1, ar_1], [al_2, ar_2], \ldots, [al_n, ar_n]$ and $[bl_1, br_1], [bl_2, br_2], \ldots, [bl_n, br_n]$.

Initially, all segments $[al_i, ar_i]$ are equal to $[l_1, r_1]$ and all segments $[bl_i, br_i]$ are equal to $[l_2, r_2]$.

In one step, you can choose one segment (either from the first or from the second list) and extend it by 1. In other words, suppose you've chosen segment $[x, y]$ then you can transform it either into $[x - 1, y]$ or into $[x, y + 1]$.

Let's define a total intersection I as the sum of lengths of intersections of the corresponding pairs of segments, i.e. \sum\limits_{i=1}^{n} {\text{intersection_length}([al_i, ar_i], [bl_i, br_i])}. Empty intersection ha...

length 0 and length of a segment $[x, y]$ is equal to $y - x$.

What is the minimum number of steps you need to make I greater or equal to k?

Input

The first line contains the single integer t ($1 \le t \le 1000$) — the number of test cases.

The first line of each test case contains two integers n and k ($1 \le n \le 2 \cdot 10^5$; $1 \le k \le 10^9$) — the length of lists and the minimum required total intersection.

The second line of each test case contains two integers l_1 and r_1 ($1 \le l_1 \le r_1 \le 10^9$) — the segment all $[al_i, ar_i]$ are equal to initially.

The third line of each test case contains two integers l_2 and r_2 ($1 \le l_2 \le r_2 \le 10^9$) — the segment all $[bl_i, br_i]$ are equal to initially.

It's guaranteed that the sum of n doesn't exceed $2 \cdot 10^5$.

Output

Print t integers — one per test case. For each test case, print the minimum number of step you need to make I greater or equal to k.

Example

Input
3 3 5 1 2 3 4 2 1000000000 1 1 999999999 999999999 10 3 5 10 7 8

Output
7 2000000000 0

Note

In the first test case, we can achieve total intersection 5, for example, using next strategy:

- make $[al_1, ar_1]$ from $[1, 2]$ to $[1, 4]$ in 2 steps;
- make $[al_2, ar_2]$ from $[1, 2]$ to $[1, 3]$ in 1 step;
- make $[bl_1, br_1]$ from $[3, 4]$ to $[1, 4]$ in 2 steps;
- make $[bl_2, br_2]$ from $[3, 4]$ to $[1, 4]$ in 2 steps.

In result, I = \text{intersection_length}([al_1, ar_1], [bl_1, br_1]) + \text{intersection_length}([al_2, ar_2], [bl_2, br_2]) + \\ + \text{intersection_length}([al_3, ar_3], [bl_3, br_3]) = 3 + 2 + 0 = 5

In the second test case, we can make $[al_1, ar_1] = [0, 1000000000]$ in 1000000000 steps and $[bl_1, br_1] = [0, 1000000000]$ in 1000000000 steps.

In the third test case, the total intersection I is already equal to $10 > 3$, so we don't