Problem C. Gifts Fixing

Time limit 1000 ms
Mem limit 262144 kB

You have n gifts and you want to give all of them to children. Of course, you don't want to offend anyone, so all gifts should be equal between each other. The i-th gift consists of a_i candies and b_i oranges.

During one move, you can choose some gift $1 \le i \le n$ and do one of the following operations:

- eat exactly **one candy** from this gift (decrease a_i by one);
- eat exactly **one orange** from this gift (decrease b_i by one);
- eat exactly **one candy** and exactly **one orange** from this gift (decrease both a_i and b_i by one).

Of course, you can not eat a candy or orange if it's not present in the gift (so neither a_i nor b_i can become less than zero).

As said above, all gifts should be equal. This means that after some sequence of moves the following two conditions should be satisfied: $a_1 = a_2 = \dots = a_n$ and $b_1 = b_2 = \dots = b_n$ (and a_i equals b_i is **not necessary**).

Your task is to find the **minimum** number of moves required to equalize all the given gifts.

You have to answer t independent test cases.

Input

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

The first line of the test case contains one integer n ($1 \le n \le 50$) — the number of gifts. The second line of the test case contains n integers $a_1, a_2, \dots, a_n$ ($1 \le a_i \le 10^9$), where a_i is the number of candies in the i-th gift. The third line of the test case contains n integers $b_1, b_2, \dots, b_n$ ($1 \le b_i \le 10^9$), where b_i is the number of oranges in the i-th gift.

Output

For each test case, print one integer: the **minimum** number of moves required to equalize all the given gifts.

Sample 1

Input	Output

Input	Output
5 3 3 5 6 3 2 3 5 1 2 3 4 5 5 4 3 2 1 3 1 1 1 2 2 2 6 1 1000000000 1000000000 1000000000 1000000000 1000000000 1 1 1 1 1 1 3 10 12 8 7 5 4	6 16 0 4999999995 7

Note

In the first test case of the example, we can perform the following sequence of moves:

- choose the first gift and eat one orange from it, so $a = [3, 5, 6]$ and $b = [2, 2, 3]$;
- choose the second gift and eat one candy from it, so $a = [3, 4, 6]$ and $b = [2, 2, 3]$;
- choose the second gift and eat one candy from it, so $a = [3, 3, 6]$ and $b = [2, 2, 3]$;
- choose the third gift and eat one candy and one orange from it, so $a = [3, 3, 5]$ and $b = [2, 2, 2]$;
- choose the third gift and eat one candy from it, so $a = [3, 3, 4]$ and $b = [2, 2, 2]$;
- choose the third gift and eat one candy from it, so $a = [3, 3, 3]$ and $b = [2, 2, 2]$.