

Pairing Cards 2

Time and memory limits: 1 second, 256 MB

You have found two unusual decks of N cards each. Each card has an integer value written on it. The i th card in the first deck has the value A_i , and the i th card in the second deck has the value B_i . The values in each deck are in nondecreasing order.

Your task is to rearrange the $2N$ cards into N pairs, such that each pair contains one card from each deck and each card belongs to exactly one pair. Before forming the pairs, you must choose two integers S and D . Every pair must satisfy **at least one** of the following conditions, using these same values of S and D :

- The sum of their values is equal to S . That is, $A_i + B_j = S$. Or,
- The value from the first deck minus the value from the second deck is equal to D . That is, $A_i - B_j = D$.

The same values of S and D must be used for every pair. You may choose a negative value for D . The difference is **not** an absolute difference.

Determine whether it is possible to choose S and D and divide the cards into N valid pairs according to the rules above.

Subtasks and constraints

Your program will be graded using many secret tests. Every test follows some rules:

- $1 \leq N \leq 1\,000\,000$.
- $1 \leq A_i, B_i \leq 1\,000\,000\,000$ for all i .
- $A_i \leq A_{i+1}$ and $B_i \leq B_{i+1}$ for all $1 \leq i < N$. That is, the values in each deck are in nondecreasing order. Values may be repeated.

The secret tests are divided into subtasks. Your program must correctly solve **every test** within a subtask to earn the marks for that subtask:

- For Subtask 1 (20 marks), $N \leq 100$.
- For Subtask 2 (20 marks), $N \leq 1000$.
- For Subtask 3 (35 marks), $A_N - A_1 > B_N - B_1$.
- For Subtask 4 (25 marks), no special rules apply.

Input

Your solution must read input and print output. We recommend using the solution templates (which you can find on the competition website) to help you with input and output.

The input follows a specific format:

- The 1st line contains the integer N .
- The 2nd line contains N integers describing the values in the first deck. The i th of these is A_i .
- The 3rd line contains N integers describing the values in the second deck. The i th of these is B_i .

Output

Your program must output YES if it is possible to choose S and D and divide the cards into N valid pairs, and NO otherwise.

Sample input 1

```
4
1 3 7 9
6 6 8 8
```

Sample output 1

YES

Sample input 2

```
3
1 1 1
1 2 3
```

Sample output 2

NO

Sample input 3

```
3
1 4 8
3 6 10
```

Sample output 3

YES

Explanation

- In the 1st sample case, you can choose $S = 9$ and $D = 1$. Pair the cards with values $(1, 8)$, $(3, 6)$, $(7, 6)$ and $(9, 8)$. The first two pairs have a sum of $S = 9$, and the last two pairs have a difference of $D = 1$.
- In the 2nd sample case, every card in the first deck has value 1. For any chosen S and D , these cards can only be paired with cards of value $S - 1$ or $1 - D$. The second deck contains three different values, so it is impossible to pair every card.
- In the 3rd sample case, you can choose $S = 0$ and $D = -2$. Pair the cards with values $(1, 3)$, $(4, 6)$ and $(8, 10)$. Every pair has a difference of $D = -2$.