Permuting Two Arrays

Consider two n-element arrays of integers, $A = [A[0], A[1], \ldots, A[n-1]$ and $B = [B[0], B[1], \ldots, B[n-1]]$. You want to permute them into some A' and B' such that the relation $A'[i] + B'[i] \geq k$ holds for all i where $0 \leq i < n$. For example, if $A = [0, 1]$, $B = [0, 2]$, and $k = 1$, a valid A', B' satisfying our relation would be $A' = [1, 0]$ and $B' = [0, 2]$, $1 + 0 \geq 1$ and $0 + 2 \geq 1$.

You are given q queries consisting of A, B, and k. For each query, print YES on a new line if some permutation A', B' satisfying the relation above exists. Otherwise, print NO.

Function Description

Complete the *twoArrays* function in the editor below. It should return a string, either YES or NO.

twoArrays has the following parameter(s):

- *k*: an integer
- *A*: an array of integers
- *B*: an array of integers

Input Format

The first line contains an integer q, the number of queries.

The next q sets of 3 lines are as follows:

- The first line contains two space-separated integers n and k, the size of both arrays A and B, and the relation variable.
- The second line contains n space-separated integers $A[i]$.
- The third line contains n space-separated integers $B[i]$.

Constraints

- $1 \leq q \leq 10$
- $1 \leq n \leq 1000$
- $1 \leq k \leq 10^9$
- $0 \leq A[i], B[i] \leq 10^9$

Output Format

For each query, print YES on a new line if valid permutations exist. Otherwise, print NO.

Sample Input

```
2
3 10
2 1 3
7 8 9
4 5
1 2 2 1
3 3 3 4
```

Sample Output

```
YES
NO
```

Explanation

We perform the following two queries:

1. $A = [2, 1, 3]$, $B = [7, 8, 9]$, and $k = 10$. We permute these into $A' = [1, 2, 3]$ and $B' = [9, 8, 7]$ so that the following statements are true:

   - $A[0] + B[1] = 1 + 9 = 10 \geq k$
   - $A[1] + B[1] = 2 + 8 = 10 \geq k$
   - $A[2] + B[2] = 3 + 7 = 10 \geq k$

   Thus, we print `YES` on a new line.

2. $A = [1, 2, 2, 1]$, $B = [3, 3, 3, 4]$, and $k = 5$. To permute A and B into a valid A' and B', we would need at least three numbers in A to be greater than 1; as this is not the case, we print `NO` on a new line.