Pac-Man (pacman)

Alessandro is coding his new 3D version of Pac-Man. The game board is described by a three-dimensional grid, some cells are blocked and the others are free. Pac-Man and the ghosts can move to any free cell sharing a face with the current cell.

Alessandro instructed ghosts to move in a very simple way. If a ghost wants to move from cell A to cell B, it will repeat the following procedure until it reaches its destination or fails:

- If the ghost can decrease the distance along the x axis (i.e. $|A_x - B_x|$) it will do so by moving one cell along the x axis;
- otherwise if it can decrease the distance along the y axis (i.e. $|A_y - B_y|$) it will do so by moving one cell along the y axis;
- otherwise if it can decrease the distance along the z axis (i.e. $|A_z - B_z|$) it will do so by moving one cell along the z axis;
- otherwise it fails.

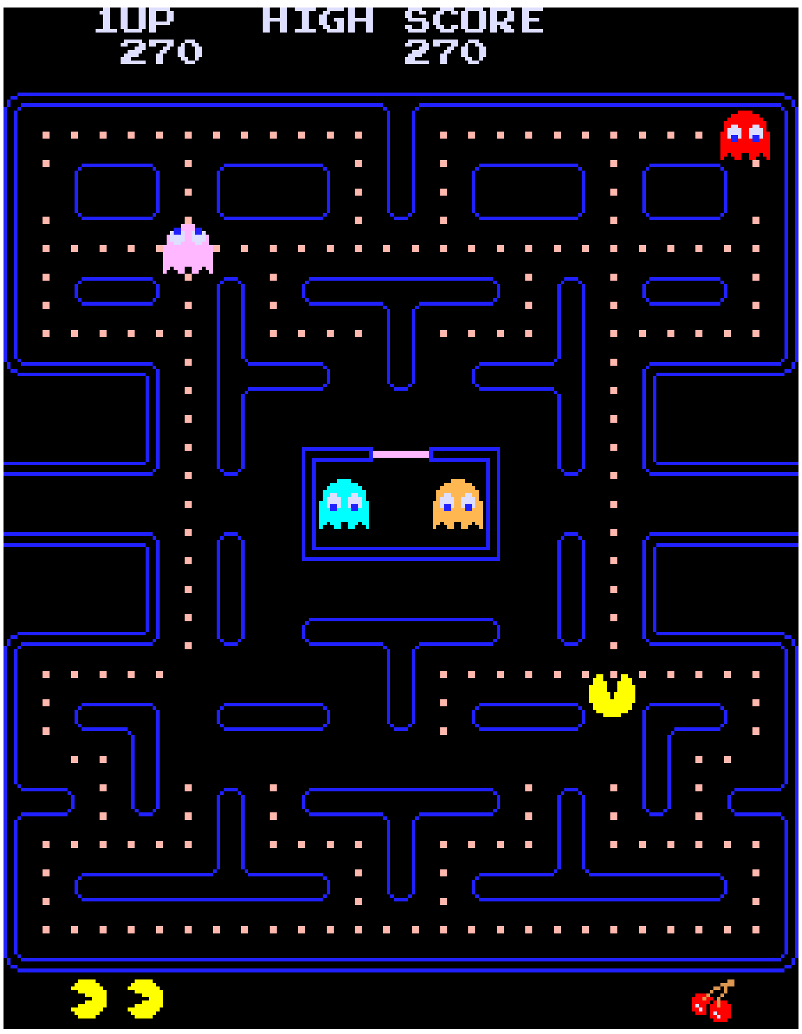

Figure 1: Alessandro playing a classic game of Pac-Man.

Alessandro has three arrays X_i, Y_i and Z_i indexed from 0 to $N-1$ that describe the coordinates of the free cells. He is wondering whether the strategy above is smart enough to control the ghosts.

Help him by writing a program that determines whether for every pair of cells A and B a ghost will succeed in reaching cell B starting from cell A!

☞ Among the attachments of this task you may find a template file `pacman.*` with a sample incomplete implementation.

Input

The input file consists of:

- a line containing integer N.
- a line containing the N integers $X_0, \ldots, X_{N-1}$.
- a line containing the N integers $Y_0, \ldots, Y_{N-1}$.
- a line containing the N integers $Z_0, \ldots, Z_{N-1}$.

Output

The output file must contain a single line consisting of `YES` if ghosts will always succeed in reaching their destinations or `NO` otherwise.

Constraints

- $1 \leq N \leq 100\,000$.
- $0 \leq X_i, Y_i, Z_i < 100\,000$ for each $i = 0 \ldots N-1$.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

– **Subtask 1** (0 points)	Examples.
– **Subtask 2** (18 points)	$N \leq 100$ and $X_i, Y_i, Z_i < 100$.
– **Subtask 3** (19 points)	$N \leq 7500$ and $X_i, Y_i, Z_i < 100$.
– **Subtask 4** (24 points)	$N \leq 1000$ and $Z_i = 0$.
– **Subtask 5** (22 points)	$X_i, Y_i, Z_i < 100$.
– **Subtask 6** (17 points)	No additional limitations.

Examples

input	output

4
0 0 1 1
0 1 1 2
0 0 0 0
	YES

8
0 1 2 2 2 1 0 0
0 0 0 1 1 1 1 0
0 0 0 0 1 1 1 1
	NO

5
0 0 1 1 2
0 1 1 0 2
0 0 0 0 2
	NO

Explanation

In the **first sample case**, ghosts can always reach their destination. For example, a ghost can move from cell $A = (0, 0, 0)$ to cell $B = (1, 2, 0)$ following path $(0, 0, 0) \rightarrow (0, 1, 0) \rightarrow (1, 1, 0) \rightarrow (1, 2, 0)$.

In the **second sample case**, ghost can't move from cell $A = (1, 0, 0)$ to cell $B = (1, 1, 1)$.