

## D. Restore Cube

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Peter had a cube with non-zero length of a side. He put the cube into three-dimensional space in such a way that its vertices lay at integer points (it is possible that the cube's sides are not parallel to the coordinate axes). Then he took a piece of paper and wrote down eight lines, each containing three integers — coordinates of cube's vertex (a single line contains coordinates of a single vertex, each vertex is written exactly once), put the paper on the table and left. While Peter was away, his little brother Nick decided to play with the numbers on the paper. In one operation Nick could swap some numbers **inside a single line** (Nick didn't swap numbers from distinct lines). Nick could have performed any number of such operations.

When Peter returned and found out about Nick's mischief, he started recollecting the original coordinates. Help Peter restore the original position of the points or else state that this is impossible and the numbers were initially recorded incorrectly.

### Input

Each of the eight lines contains three space-separated integers — the numbers written on the piece of paper after Nick's mischief. All numbers do not exceed  $10^6$  in their absolute value.

### Output

If there is a way to restore the cube, then print in the first line "YES". In each of the next eight lines print three integers — the restored coordinates of the points. The numbers in the  $i$ -th output line must be a permutation of the numbers in  $i$ -th input line. The numbers should represent the vertices of a cube with non-zero length of a side. If there are multiple possible ways, print any of them.

If there is no valid way, print "NO" (without the quotes) in the first line. Do not print anything else.

### Examples

input
0 0 0 0 0 1 0 0 1 0 0 1 0 0 1 0 1 1 0 1 1 0 1 1 1 1 1
output
YES 0 0 0 0 0 1 0 1 0 1 0 0 0 1 1 1 0 1 1 1 0 1 1 1

input
0 0 0 0 0 0 0 0 0 0 0 0 1 1 1

1 1 1  
1 1 1  
1 1 1

output

NO