Problem G. 3D Knight

Input file:	`knight.in`
Output file:	`knight.out`
Time limit:	2 seconds
Memory limit:	256 megabytes

The travelling knight problem is the famous problem that is used to show various aspects of backtracking when teaching programming. But the classic 2-dimensional version of this problem is already well studied, so let us move to 3D version of the problem.

You are given an $a \times b \times c$ parallelepiped. A chess knight can reside in any unit cube of it. The goal is to put the knight at one of the corner cubes of the parallelepiped, and find its path that visits each unit cube exactly once.

Let us denote unit cubes by three coordinates, ranging from 1 to a, 1 to b and 1 to c, respectively. The knight can move from a cube to another cube if one coordinate of the destination cube is the same, one differs by exactly one, and one differs by exactly two (the standard chess knight move). The path must start at a cube $(1, 1, 1)$.

Input

The input file contains three integer numbers a, b and c ($1 \le a, b, c \le 5$).

Output

If the required path exists, output "`YES`" at the first line of the output file. After that list the cubes in order the are visited along the path — abc lines, each of them must contain three numbers — the coordinates of the corresponding cube.

If there is no required path, output "`NO`" at the first line of the output file.

Example

knight.in	knight.out
4 3 2	YES 1 1 1 3 2 1 1 3 1 2 1 1 4 1 2 2 2 2 4 2 1 2 3 1 3 1 1 1 2 1 3 3 1 4 1 1 4 3 2 3 1 2 1 2 2 3 3 2 2 1 2 1 3 2 3 2 2 1 1 2 2 3 2 4 3 1 2 2 1 4 2 2

The coordinates in the example are printed two triples on a line to save space. You should print one triple on a line.