PROBLEM C: 3D-CHESS

Input file: standard input

Output file: standard output

Problem

It is year 2213 and you travel aboard a spacecraft. It was back in XXII century when rapidly developed computers, despite incredibly big number of possible combinations, definitely solved the chess problem. It became possible to predict the result when two good players sat at the opposite sides of the chessboard. Anyway for mankind it was a challenge. On one hand traditional chess game bored people, but on the other hand none wanted to leave aside a game which used to enjoy people for centuries. It raised the need to modify the rules of the game. In 2206 work started on game called 3D-Chess. As it may be easily figured out the chessboard was enhanced to third dimension (can it be still called "board"?). It is unfortunately not yet done and the work still continues on optimal size of the "chess-space" and the set of chessmen. It seems however that it will end up with more familiar size 8*8*8, but to stress it once again - it is not yet decided. One of the most keen on the chess players was captain Picard who also worked on optimal set of chessmen. His invention was a "superknight". A superknight slightly resembles its ancestor, but it was also enhanced to 3D space. Thus it can move in one of the following ways:

- two fields forward (in any direction) and one sidewise (all within one plane) - like a "regular" knight.
- two planes upwards or downwards and one sidewise in one of the four possible dimensions
- teleportation - movement to field which is a mirror reflection of the standpoint with respect to one of the three axes of symmetry of the "chess-space" parallel to its edges.

For superknight it is not relevant whether the fields between the starting field and ending one are occupied or not (again like for the "regular" knight).
It only matters whether the ending field is empty. Then the movement can be performed.
There has to be a restriction among so many capabilities of the superknight. It cannot perform the same sort of movements consecutively (just not to fall into routine).
As an example in the table below it is shown what movements are allowed for a superknight standing on the field with coordinates (2, 3, 4). The "chess-space" has size 8*8*8 and the numbers of the fields start form (1, 1, 1).

(4, 4, 4)	(3, 3, 6)	(7, 3, 4)
(4, 2, 4)	(1, 3, 6)	(2, 6, 4)
(3, 5, 4)	(2, 4, 6)	(2, 3, 5)
(3, 1, 4)	(2, 2, 6)	
(1, 5, 4)	(3, 3, 2)	
(1, 1, 4)	(1, 3, 2)	
(2, 4, 2)		
(2, 2, 2)		

Your task is to write a program which could help poor captain Picard in his research. It should calculate the minimal number of movements necessary to get the knight from one field to the given another.

Input

It can contain several sets of data. Each one starts with a line with a number N being size of the "chess-space" (its a cube N*N*N). Successive two lines contain coordinates of the starting and the ending point. Successive lines contain description of obstacles in the "chess-space". Each obstacle is a rectangular parallelepiped and is described in the following way: first three numbers stand for one vertex coordinates, successive three numbers stand for sizes of its edges. Line containing numbers: 0, 0, 0, 0, 0, 0 ends the obstacle description. It may be assumed that $3 \leq N \leq 16$.

Output

For each set of data you should generate one line containing the minimal number of movements. If the solution did not exist then you should output a string "The solution doesn't exist"

Example

Input file

```
8
1 1 1
2 3 1
0 0 0 0 0 0
8
1 1 1
8 8 8
0 0 0 0 0 0
16
2 3 4
8 8 8
1 1 1 8 1 1
3 5 6 2 2 2
3 4 4 3 3 1
1 1 4 1 2 1
7 6 8 1 1 1
6 7 8 1 1 1
7 8 6 1 1 1
8 7 6 1 1 1
8 8 1 1 1 1
8 1 8 1 1 1
1 8 8 1 1 1
0 0 0 0 0 0
0
```

Output file

```
1
5
7
```