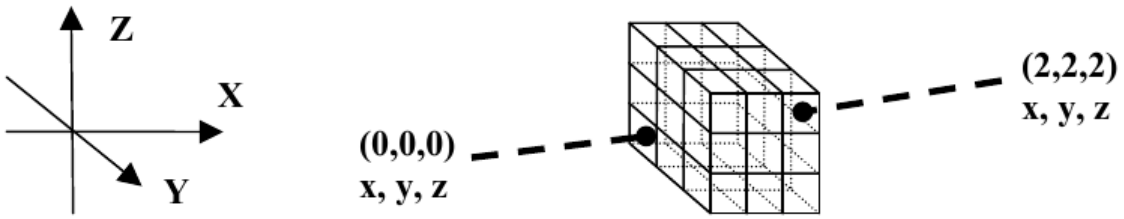
2805 Dominatrix

Dominatrix is a game that uses 3-D domino pieces. The pieces are composed of blocks that fit in a $3 \times 3 \times 3$ matrix, and the facets of each block are either blank, or marked with a 1-, 2-, 3-, 4-, 5- or 6-dot configuration. You are to create a program that determines if 2 Dominatrix pieces are the same. In other words, if one piece can be rotated and/or translated within the $3 \times 3 \times 3$ matrix, the two pieces are identical.

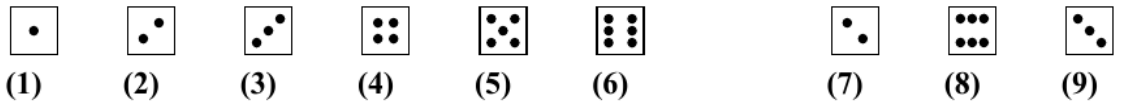
The data for 1 piece is encoded in the following manner: First, a series of 0's and 1's determines where a block exists in the $3 \times 3 \times 3$ matrix. The blocks are laid out along the X-axis, Y-axis, and Z-axis. Representing the blocks in (x,y,z) coordinates, they are laid out $(0,0,0), (0,0,1), (0,0,2, (0,1,0), \ldots (2,2,2)$.



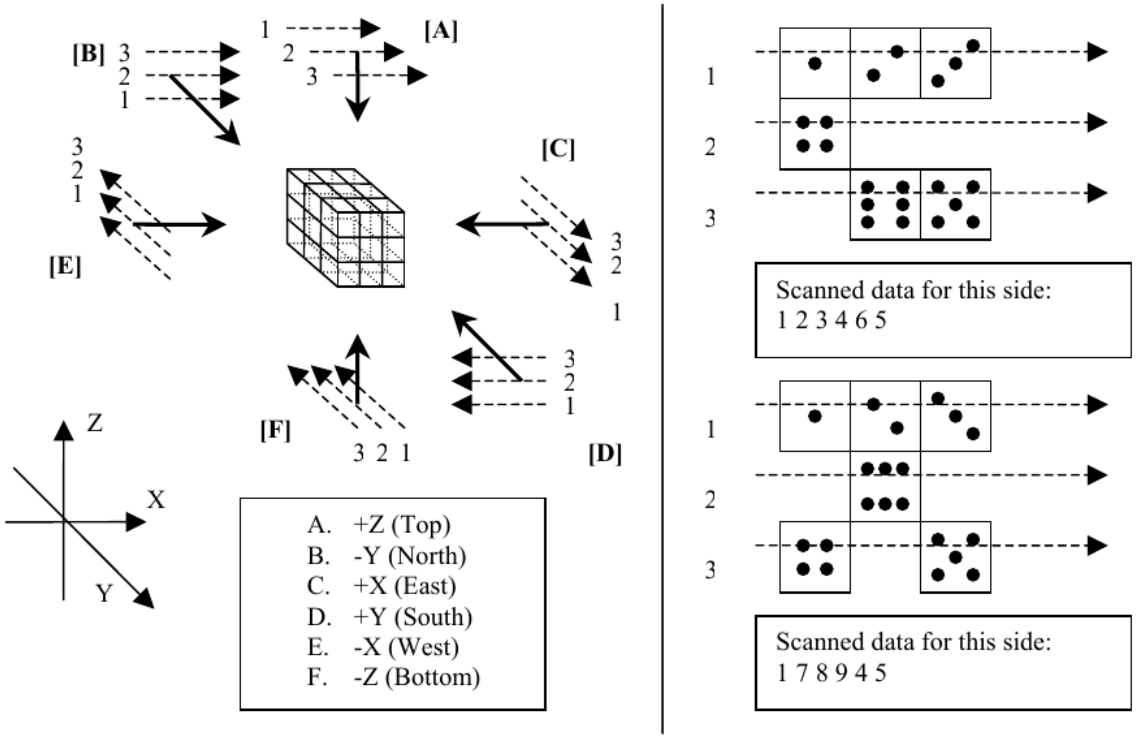
The succeeding data is derived from scanning the dot configurations on each visible facet of each block, from each of the 6 sides of the $3 \times 3 \times 3$ matrix containing the Dominatrix piece. The scanner scans first from the +Z (top) side of the matrix, then swivels down to the -Y (north) side, then round to the +X (east) side, then to the +Y (south) side, then to the -X (west) side, then finally to the -Z (bottom) side.

When the scanner reads each side of the $3 \times 3 \times 3$ matrix, it scans from left to right, then from top to bottom. The number of dots for each facet for each block is encoded into the file. If there are less than 9 facets scanned for each side of the matrix because some blocks are missing, only the dot configurations that are read are encoded.

Facets of blocks that cannot be reached by the scanner from any side are assumed to be blank.



There are 9 dot configurations that can be scanned from each block facet, since there are 2 possible orientations for the 2-, 3-, and 6-dot facets. The alternate configurations for these are encoded as 7, 8 and 9 respectively.



Input

The input is composed of at most 10 test cases, each test case being 2 consecutive sequences of data for 2

Dominatrix pieces. The line following the last test case contains '`#`'.

For each test case, first a sequence of 27 numbers, composed of '`0`'s/'`1`'s, 1 for the existence of blocks at coordinates $(0,0,0),(0,0,1),(0,0,2),(0,1,0),\ldots(2,2,2)$ is inputted. Following is a sequence of data representing the scanning of each side of the piece as described above. After the sequence of data for the 1st Dominatrix piece, the sequence of data for the 2nd Dominatrix piece follows after a carriage return.

All numbers are delimited by any number of spaces and/or carriage returns.

Output

For each test case, the program should output a single line, containing '`EQUAL`' if the 2 pieces are equal, and '`NOT EQUAL`' if they are not.

Sample Input

```
1 1 1 1 1 1 1 1 1
1 1 1 1 0 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2
3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4
5 5 5 5 5 5 5 5 5 6 6 6 6 6 6 6 6 6
1 1 1 1 1 1 1 1 1
1 1 1 1 0 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 5 5 5 5 5 5 5 5 5
```

```
2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3
4 4 4 4 4 4 4 4 9 9 9 9 9 9 9 9
#
```

Sample Output

```
EQUAL
```