



### PROBLEM 9 – POLYCUBES

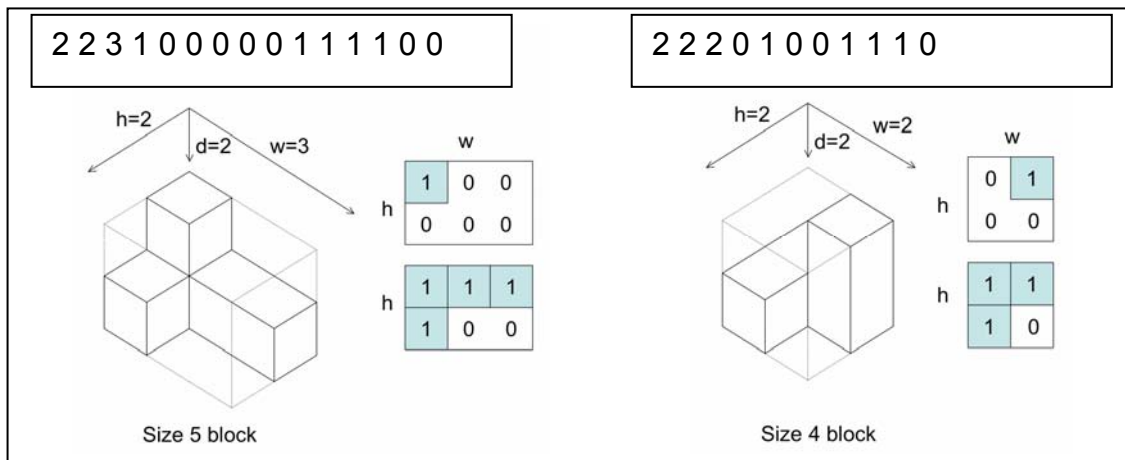
A Grant's cube is a  $3 \times 3 \times 3$  cube made of (usually wooden) *blocks*, also known as *polycubes*. Each of the blocks is made up of a number of unit cubes, joined on faces. In the standard Grant's puzzle there are 3 *tetracubes* (blocks made of 4 units), and 3 *pentacubes* (blocks made of 5 units), making a total volume of 27 units. However, larger or smaller blocks are allowed in this problem.

Your task is to determine if a *candidate set of blocks* can fit together to form a Grant's cube. Finding a solution may require rotation of some of the blocks from their initial orientations, and it is possible that no solution exists for a given set of blocks (that they can't form a cube at all). It does not matter whether the blocks can be physically brought together when determining if a solution exists. As long as they can be oriented so as to completely fill the  $3 \times 3 \times 3$  space without overlapping, that is enough for this problem.

#### INPUT FORMAT

The input file describes one or more scenarios. Each scenario is described on  $k+1$  lines, where  $k$  is the number of blocks forming a candidate set. The first line contains a label, which is a string of 1 to 5 digits, and the number  $k$  of blocks in the set,  $1 \leq k \leq 6$ , separated by single spaces.

Each of the next  $k$  lines encodes a single block. Each such line begins with three dimensions,  $d, h, w$ ,  $1 \leq d, h, w \leq 3$ , representing the smallest rectangular solid that the block fits into, followed by a list of  $d \times h \times w$  binary digits, all items separated by single spaces. These binary digits are logically arranged in  $d$  groups of  $h$  groups of  $w$  integers, where a 1-digit means that the corresponding cell in the rectangular solid is part of the encoded block, whereas a 0-digit means that the corresponding cell is not part of the block. For an explanatory example, see the following diagram, which describes two blocks and their encodings:



The input will be terminated by a line consisting of two zeros (0), separated by a single space. This line should not be processed.



SAMPLE INPUT:

```
01 6
2 2 3 1 0 0 0 0 1 1 1 1 0 0
2 2 3 1 0 0 0 0 1 1 1 0 0 1
2 2 3 0 1 1 0 0 1 1 0 1 0 0
2 2 2 0 1 0 0 1 1 1 0
2 2 2 0 0 1 0 1 1 1 0
2 2 2 1 0 0 0 1 1 1 0
001 3
1 3 3 1 1 1 1 1 1 1 1
3 1 3 1 1 1 1 1 1 1 1
3 3 1 1 1 1 1 1 1 1 1
002 3
1 3 3 1 1 1 1 1 1 1 1
1 3 3 1 1 1 1 1 1 0 0
2 3 3 1 1 1 1 1 1 1 0 0 0 0 0 0 1 1
003 3
1 3 3 1 1 1 1 1 1 1 1
1 3 3 1 1 1 1 1 1 0 0
2 3 3 1 1 1 1 1 1 1 0 0 0 0 0 1 0 1
0 0
```

OUTPUT FORMAT

For each scenario, i.e., for each input candidate block set, the program should output a single line of text. The line of text should consist of the scenario label, followed the word "YES" (in uppercase) if there is at least a solution, otherwise "NO" (in uppercase), separated by a single space.

SAMPLE OUTPUT:

```
01 YES
001 YES
002 YES
003 NO
```