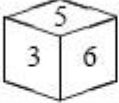
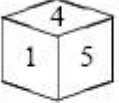
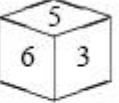
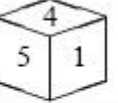
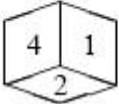
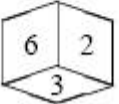
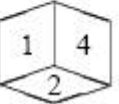
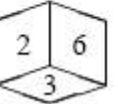


The following picture shows four dice. Or are they all dice?

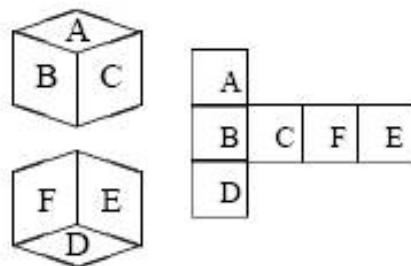
	(a)	(b)	(c)	(d)
front-top				
back-bottom				

Artifacts (a) and (b) are dice; (c) and (d) are not dice, but they are the same — you can see that by a suitable rotation of (d). Thus, here we have two classes of dice-like artifacts (diceoids), classified by comparison under suitable rotations.

You are given a bag of diceoids, and your job is to classify them and report how many classes there are. (Your job is not to determine how many are dice.) To facilitate description of diceoids in a text medium, we label the faces of the cube:

then a diceoid is encoded as a sequence of six numbers d_A, \dots, d_F , meaning that face i bears d_i dots.

Example: 5 3 6 2 1 4 means that face A has 5 dots, face B has 3 dots, face C has 6 dots, etc. It is entirely possible that different faces have the same number of dots, e.g., 2 2 2 3 3 3, but every face has at least 1 dot and at most 6 dots.



Input

The input file contains several test cases. The description of each test case is given below:

Each test case begins with the number of diceoids n ($n \leq 1000$), followed by the description of n diceoids.

Output

For each test case your program should output the number of classes. Follow the format specified in the output for sample input.

Sample Input

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

Sample Output

```
4
```