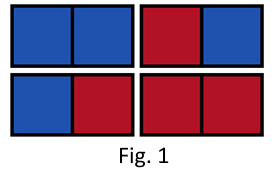
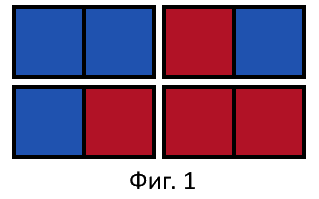
Sashka won’t stop playing with her favourite dominoes. She has four types of dominoes (Fig. 1). They have two sides, left and right respectively. Both of the sides are colored in blue or red. For convenience, Sashka denoted them as follows:

* Domino №1: blue-blue
* Domino №2: blue-red
* Domino №3: red-blue
* Domino №4: red-red

You should note that blue-red is different from red-blue.

Sashka takes out all the dominoes she has and orders them in a sequence. Sashkadenotes a sequence of dominoes as *beautiful* if each adjacent side of the dominoes is with different color. More precisely, if the left side of the -th domino is and the right is , then for every , . A sample *beautiful* sequence is shown below (Fig. 2).



In Bulgaria there are domino sets, where the -th set contains dominoes from the first type, from the second type, from the third type and from the forth type. At first, Sashka took the *intermediate* assessment of a domino set to be the count of different *beautiful* sequences, which can be made from it. Two domino sequences are different if two different dominoes are on the same positions in the sequences. She realized that the count of such sequences could be pretty big, so she changed the method of assessment. The new *intermediate* assessment of a domino set is taken to be the count of achievable *beautiful* sequences modulo . The bigger the remainder is, the better the *intermediate* assessment is for the set, but there is more. Sashka is allowed to recolor dominoes in the -th set to increase her assessment of the set. Recoloring of a domino in the -th set goes like this: Sashka chooses two types of dominoes and (, , ) and decreases by and increases by (, where is the sign for assignment). **So in the end, the *final* assessment of a domino set is the maximal *intermediate* assessment of all possible recolorings**. Sashka wants to find the *final* assessment of all dominoes set in Bulgaria, but it’ll be hard for her to calculate it by hand. That’s why she asks you, as her third cousin, to write a program dominoes.cpp, which will find it.

**Input**

On the first line of dominoes.in the number is given – the number of domino sets in Bulgaria*.* The -th of the next rows contains natural numbers, ,respectively.

**Output**

For each of the domino sets, you should print a number in dominoes.out, which is its *final* assessment.

**Constraints**

For every ,

**Time Limit: 1 sec.**

**Memory Limit: 256 MB.**

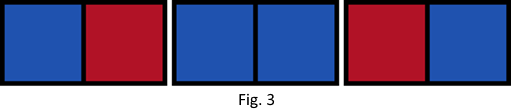
**Sample testcases**

|  |  |
| --- | --- |
| **Input (dominoes.in)** | **Output (dominoes.out)** |
| 3  1 1 1 0 0  1 0 1 1 0  3 0 0 0 2 | 1  3  3 |
| 3  3 3 3 1 3  8 1 1 0 4  10 0 0 0 5 | 120  40  5 |
| 2  9 2 5 10 5  9 8 4 5 3 | 3171168  2944656 |
| 3  30 30 30 30 20  30 60 15 15 20  60 60 0 0 20 | 999079969  997861536  0 |

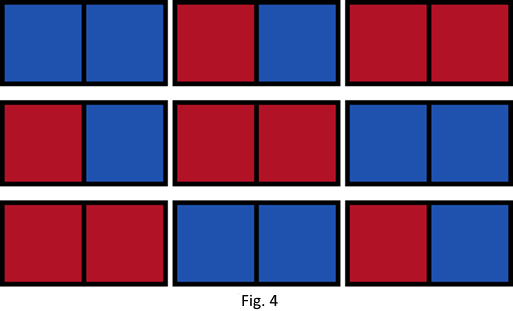
**Explanation of the sample testcases**

For the first testcase:

* For the first domino set the only possible sequence is:



* For the second domino set, the three possible sequences are the following:



* For the third domino set,

The rest of the sample testcases probably have many beautiful sequences, but the space won’t be enough ☺.