At random

[Due to SPOJ restrictions, this problem has been modified with respect to the original version used in the Argentinian Programming Tournament of 2014 in order to have multiple test cases per input file. The original version of this problem (in Spanish) can be found at http://dc.uba.ar/events/icpc/download/problems/tap2014-problems.pdf]

There are many and very different card games, their origin going back to ancestral times. Sometimes it can be surprising that they continue to provide entertainment after so many centuries of being played with the same rules, but then we should remember that each game is essentially different from all the other ones played in the history of humankind, given the huge amount of possible ways to order the cards before the beginning of each game. Indeed, few games are any fun if we always use the cards in the same order, or if there is a correlation between successive cards that allows us to guess the order in which they will appear. This is the reason why it is customary to shuffle the cards before starting each game, and for this same reason we ask you to make a program to check that a sequence of cards has been properly shuffled.

To simplify the problem, we will concentrate only on decks of *Spanish cards*, which consist of **48** different playing cards. Each card is identified by a *value*, which is a number from **1** to **12**, and a *suit*, which can be "*clubs*", "*cups*", "*golds*" or "*swords*". However, because we don't want to excessively simplify your task, we will take into account that not all games use all **48** cards of the deck. Given a sequence of **N** cards, we say it has been *properly shuffled* if there are no two successive cards sharing the same value or suit. Otherwise, we say the deck has not been properly shuffled. Can you help us decide if a given sequence has been properly shuffled?

Input

The first line contains an integer number **T**, the number of test cases (**1 ≤ T ≤ 200**). **T** test cases follow.

The first line of each test case contains an integer **N**, representing the number of cards that are used in the game we are considering (**2 ≤ N ≤ 48**). Each of the following **N** lines contains the description of a card in the sequence we want to analyze, given by an integer **V** representing its value (**1 ≤ V ≤ 12**) and a character **P** representing its suit: **'b'** for clubs, **'c'** for cups, **'o'** for golds and **'e'** for swords. All the cards in the input are different, and they are given in the input in the same order they appear in the sequence.

Output

For each test case, print a single line containing a character indicating if the sequence of cards given in the input has been properly shuffled or not. The character should be a **'B'** if it has been properly shuffled, and an **'M'** otherwise.

Example

Input:

```
4
4
1 b
2 c
3 e
4 o
3
1 b
2 b
3 c
3
1 b
1 c
2 e
32
5 c
2 b
4 e
3 o
12 b
1 c
7 e
6 c
12 e
4 o
1 b
6 o
3 e
12 o
11 e
12 c
5 o
10 b
9 o
3 c
4 b
11 c
8 e
9 c
1 e
4 c
8 b
2 o
6 b
9 e
```

```
7 b
5 e
```

Output:

```
B
M
M
B
```