M and **J** are playing a game. The description of the game is very simple. They have a few displays where several digits will be shown. There are buttons under each digit of every display. If someone presses the ith button of a display once, the value of the i-th digit of that display will increase by one. Each of these displays has one fault in common. The 0-th digit (least significant digit) can't be increased because its button is broken.

Each of these displays can be described using two parameters: L and B. L is the length of the display or the number of digits shown by the display. If L is 3 then the display can show **three** (3) digits side by side (we can consider it like a 3-digit number). B is the base of the display. It is important in two aspects. Firstly, it limits the number of times you can press a button. Secondly, the number displayed in the display will be interpreted as a B-based number with L digits. Suppose B is 8 and L is 4. Then the numbers shown by the display will be octal numbers and the length of the numbers will be 4. Also you can press each of the **three** working (not broken) buttons at most **seven** (7) times. Example of such a display is shown in Figure 1.

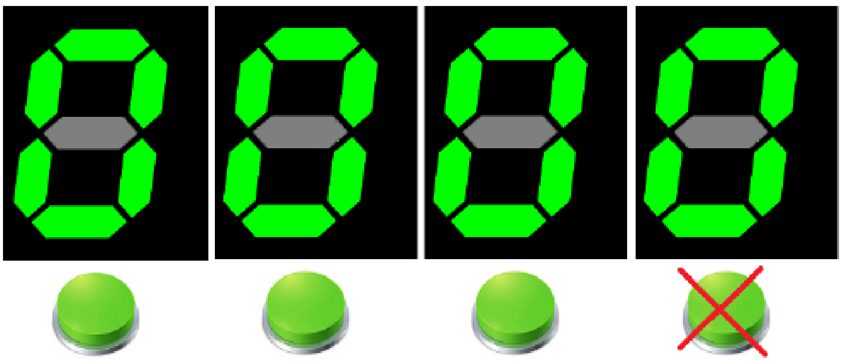

Figure 1: A display with $L = 4$ and $B = 8$

M and **J** have N of these displays. Each of the display has its own L and B. Now the game **M** and **J** are playing goes like this:

1. **M** plays first, **J** plays second. After that they alternate moves.
2. In each move, a player selects a display. Then selects a digit. And presses its button. A digit can only be selected, if it hasn't reached its limit $B - 1$ already. A display can only be selected, if there is a digit which can be selected. However since the switch for the rightmost digit is already broken, you can not choose this button for any display.
3. After a move, if the summation of numbers (After converting it to decimal base) shown in the displays is divisible by 3, then the player making that move loses and the other player wins.
4. If there is no move possible, then the game ends in draw.

Given description of N displays, you need to find out the outcome of the game. **M** and **J** both plays optimally. Initially every digit of every display is set to 0 (zero).

Input

First line of input consists of an integer T ($T \leq 100$), the number of test cases. Each test case starts with an integer N ($0 < N < 10$), the number of displays. Next N line each contains two integers, L ($0 < L < 10^9$) and B ($1 < B < 10^9$) which are the parameters that describes the i-th display.

Output

For each case print one line: '`Case` X`:` S', where X is the case number. S is either '`M`', '`J`' or '`Draw`' based on the outcome of the game in that case. There is no new-line between cases.

Explanation

Case 1: 00000 => 00010 => 01010 => 01110 => 11110 which is 30 in decimal and divisible by 3. So **J** loses and **M** wins. However this is one possible valid game sequence. But if two players play optimally **J** will always lose and thus **M** will always win.

Case 2: (000, 00) => (000, 10) Sum = 0 + 6 = 6.

(000, 00) => (100, 00) => (100, 10) => (110,10) Sum = 6 + 6 = 12.

(000, 00) => (100, 00) => (100, 10) => (100, 20) => (100, 30) => (100, 40) => (100, 50) => (110, 50) Sum = 6 + 30 = 36

In this way, in every game sequence **M** is doomed to reach such a configuration where the sum will be divisible by 3. Hence, in this scenario **J** will win and **M** will lose.

Case 3: (00) => (10) which is 2 in decimal and not divisible by 3. There is no move left. So the game ends in a draw.

Sample Input

```
3
1
5 2
2
3 2
2 6
1
2 2
```

Sample Output

```
Case 1: M
Case 2: J
Case 3: Draw
```