Flip Game

Time Limit: 1 sec

Do you know Alice and Bob? They are famous fictional characters in cryptographic context. Almost all cryptographic theories include a story about Alice and Bob explaining how they can communicate securely. Being tired of reading this complex stories about them, Alice and Bob are planning to have some relax and play a simple yet interesting flip game.

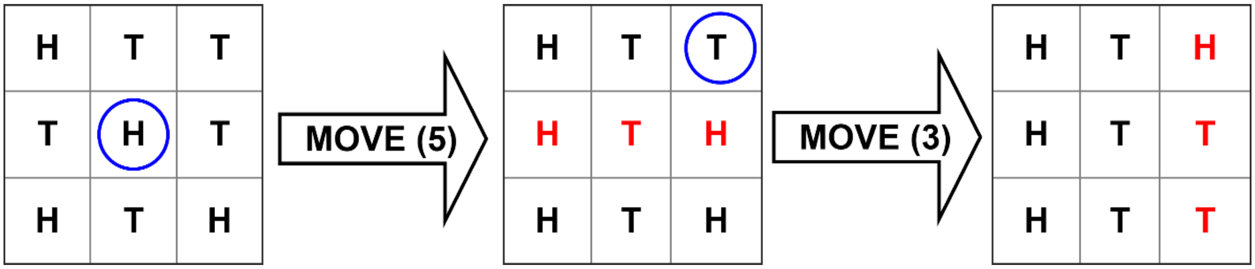
The game board can be considered as an **(N x N)** grid. Each cell of the board has an identification number from **1 to N^2** assigned in order from top-left cell to bottom-right cell. To make it clear consider a (4x4) game board:

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Each cell of the board contains a coin. Initially all the coins are **tail** faced. A move in this game is defined by a cell number. The rules for performing a move are defined as follows:

1. If the coin on the specified cell is **head** faced, then flip all the coins on the same **row** of the cell.
2. If it is **tail** faced, flip all the coins on the same **column** of the cell.

Flipping a coin simply means head-faced coin will tail-faced and tail-faced will be head-faced. Consider the following example:



H	T	T
T	H	T
H	T	H

H	T	T
H	T	H
H	T	H

H	T	H
H	T	T
H	T	T

The game is consists of **M** random moves which are generated by an unbiased random-move-generator device. These moves are performed by Alice or Bob in the given order. Since the rules for performing a move have already specified, they are not strategic. So it doesn't matter who performed the moves, Alice or Bob.

After performing the moves, if the number of head-faced coins and the number of tail-faced coins are **equal** then, it is considered as a **Draw.** If the number of **heads** is greater than the number of **tails** then **Alice** wins. Otherwise **Bob** wins the game.

As the number of moves can be large, instead of performing the moves by hand, Alice and Bob requested you to write a program to simulate the moves. You are given the size of the board and a set of moves, you have to simulate the moves in the given order and finally determine the winner.

Input:

Input starts with an integer **T (<= 100)**, denoting the number of test cases.

Each test case consists of two lines. First line contains two integers **N (1 <= N <= 10)** and **M (1 <= M <= 100)** which respectively represent the size of game board and the number of moves. Second line contains **M** space separated integers **(1 <= M_i <= N^2)** that represent the moves.

Output:

For each case, print the case number and **'Alice'** or **'Bob'** or **'Draw'** depending on the winner of the game.

Sample Input:

```
3
4 3
10 13 8
3 4
6 1 7 4
4 6
11 9 5 12 8 6
```

Sample Output:

```
Case 1: Alice
Case 2: Bob
Case 3: Draw
```