

---

# Stone Game

Input file:            **standard input**  
Output file:           **standard output**  
Time limit:            1 second  
Memory limit:         256 megabytes

Alice and Bob are always playing game! The game today is about taking out stone from the stone piles in turn.

There are  $n$  piles of stones, and the  $i$ -th pile contains  $A[i]$  stones.

As the number of stones in each pile differs from its neighbor's, they determine to take out exactly one stone from one of them in one turn without breaking that property. Alice goes first.

The player who cannot take stone will lose the game.

Now you have to determine the winner given  $n$  numbers representing the piles, if they both play optimally.

You should notice that even a pile of 0 stone is still regarded as a pile!

## Input

The first line of input file contains an integer  $T$  ( $1 \leq T \leq 100$ ), describing the number of test cases.

Then there are  $2 \times T$  lines, with every two lines representing a test case.

The first line of each test case contains only one integer  $n$  ( $1 \leq n \leq 10^5$ ) as described above.

The second line of that contains exactly  $n$  numbers, representing the number of stone(s) in a pile in order. All these numbers are ranging in  $[0, 10^9]$ .

It is guaranteed that the sum of  $n$  in all cases does not exceed  $10^6$ .

## Output

You should output exactly  $T$  lines.

For each test case, print **Case  $d$ :** ( $d$  represents the order of the test case) first, then print the name of winner, **Alice** or **Bob**.

## Example

standard input	standard output
2	Case 1: Alice
2	Case 2: Bob
1 3	
3	
1 3 1	

## Note

Sample 1:

There is a possible stone-taking sequence:  $(1, 3) \rightarrow (1, 2) \rightarrow (0, 2) \rightarrow (0, 1)$

Here is an invalid sequence:  $(1, 3) \rightarrow (1, 2) \rightarrow (1, 1) \rightarrow (0, 1)$ . Though it has the same result as the first sequence, it breaks that property "the number of stones in each pile differs from its neighbor's".