Glass Stepping Stones

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	1024 mebibytes

Alice and Bob are playing a game on a string of letters "`L`" and "`R`" (with ASCII codes 76 and 82, respectively). They take turns, starting with Alice. On their turn, a player removes one character from the string.

If the resulting string has no substring "`LR`", Alice wins; if the resulting string has no substring "`RL`", Bob wins. However, if both events occur simultaneously, the game is declared a draw.

The previous paragraph also applies to the starting string: for example, if it contains the substring "`RL`", but does not contain the substring "`LR`", then no one will make any moves because Alice automatically wins.

What will be the outcome of the game if both play optimally? How should Alice play to achieve this outcome?

Input

The first line contains an integer T, the number of test cases ($1 \le T \le 10^6$). The following T lines describe the test cases themselves.

Each test case is represented by a string s consisting of the letters "`L`" and "`R`" ($1 \le |s| \le 10^6$).

The sum of lengths of the strings s across all test cases does not exceed 10^6.

Output

For each test case, print "`Alice`", "`Bob`" or "`Draw`" (case-insensitive) indicating either the winner of the game or that there will be no winner. On the same line, output an integer a describing any optimal move for Alice. The value a is the position of a character in string s to remove on the first turn ($1 \le a \le |s|$). The move is optimal if it does not change the outcome of the game when both players play optimally afterwards. If the game ends before the first move, print $a = 0$ instead.

Example

standard input	*standard output*
16	Draw 10
LRRRRLRRLRRRLLLRRL	Draw 0
L	Draw 0
R	Draw 0
RRRRRRRRR	Draw 0
LLLLLLLLLLLLLLL	Alice 1
LRLL	Draw 2
RLRR	Alice 2
RLRL	Bob 0
LLLLRRR	Alice 0
RL	Bob 0
LR	Bob 2
LRLR	Draw 5
LRLLR	Alice 1
LRLRL	Bob 2
LLLLLLLLLLRRRRRRRRLLLLLLRRRR	Draw 30
LLLLLLLLLLRRRRRRRRLLLLLLLLLLRRRR	