4630 Go

Go is an oriental board game played on a 19×19 grid, with two players taking turns to place black and white markers (called stones) on the grid intersections. A chain is a collection of stones defined by the following property: two stones of the same colour on the board are in the same chain if there is a path between those stones (following grid lines on the board) such that each intersection on the path is occupied by a stone of that colour. A chain is assigned the colour of the stones it contains.

The number of liberties of a chain is the number of unoccupied intersections that are immediately adjacent to stones in the chain (again, following grid lines). If at some stage, a chain is reduced to zero liberties, it is captured by the opponent.

Given representations of Go boards, you are required to output the number of liberties of each chain on the board.

Input

The first line of the input will be a positive integer, indicating how many Go boards will follow. On the next line, the representation of the first board will begin. Each board's representation consists of 19 lines, and boards will be separated by a blank line. Each line consists of 19 characters, each character denoting whether the corresponding intersection of the board is empty(`.`), occupied by a white stone (`W`), or occupied by a black stone (`B`).

Restrictions: No chain on the board will have zero liberties.

Output

For each board representation, your program should output '`Case` X`:`', where X is the number of the board under consideration (starting at 1), followed by a new line. The next line begins with the string '`Black: `' followed by a space-separated list of the number of liberties of each Black chain on the board, sorted in increasing order. The next line begins with the string '`White: `' followed by a space-separated list of the number of liberties of each White chain on the board, sorted in increasing order.

Sample Input

```
2
B..................
...................
...................
...................
...................
...................
...................
...................
...................
...................
...................
...................
...................
...................
```

```
...................
...................
...................
...................
...................

B....WWWWB.........
.....WBBBB.........
WWWWWWB............
BBBBBBB............
...................
...................
...................
...................
...................
..........BBBB.....
............BB....
.............B....
..............BW..
...............B..
...................
...................
...................
...................
...................
```

Sample Output

```
Case 1:
Black: 2
White:
Case 2:
Black: 2 3 3 13 13
White: 2 6
```