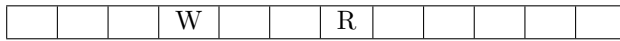
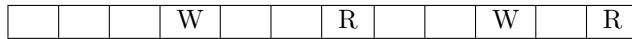


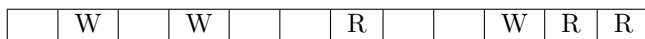
A surprisingly complex game can be played between two players on a simple one-dimensional board with up to 60 holes; each player has counters of one colour (indicated here by letters like W or R) which go into the holes. A player may move a counter anywhere on the board, as long as it does not land on or jump over any other counter. The object of the game is to block the other player so he or she has no allowable moves. In a game with one counter each, the first player can force a win on the first move by moving their counter next to the other counter. Whenever the second player tries to move away, the first player moves next to them. Eventually the second player will have no moves left. If the first player does not take this first move, then the second player can force a win. Convince yourself that these statements are true in this example:



With two counters each, the game is more complex. For example, in the following situation, if W moves first they can force a win by moving the leftmost counter one square to the right, or the rightmost counter one square to the left. Any other move (for example, moving one of the W counters next to an R counter) will mean that R can force a win. Try it and convince yourself this is true.



Even more complex situations arise with more counters; for example, draws can occur (ie neither player can force a win) as in the following case:



It is embarrassing that it is difficult to see how to win such a simple game. Please write a program so that I can play this safely by always knowing the winning first move, or, when there is no winning first move, whether the game is lost or can be drawn. Assume that a player will always win the game if it is in their power, and will always try to avoid defeat.

Input

Each line of input will be a game description, which is a string of up to 60 digits. Each digit represents a hole on the board: '0' for an empty hole, '1' for a counter belonging to the first player, or '2' for a counter belonging to the second player. Both players will have the same number of counters, and there will be at least two empty holes on the board. The input will be terminated by a line consisting of a single zero.

Output

For each input game description, one line of output must be produced. This line must be:

- '0' if the game cannot be won but can be drawn;
- '2' if the first player will lose whatever move is made;
- '1' if there is a winning first move. In this case, the '1' must be followed by a description of the winning move. The winning move must be given by two numbers, the first giving the hole in which the move begins, and the second giving the hole to which the piece moves. The holes are assumed to be numbered from left to right with the leftmost being number 1. If there is more than one winning move, choose the move starting from the smallest numbered hole. If there is more than one of these, choose whichever moves to the smallest numbered hole.

Sample Input

```
000200100000
000100200102
000010201002
000010200102
001020020001100002000
0
```

Sample Output

```
1 7 5
1 4 5
1 5 4
2
1 13 15
```