

8278 Catan's Longest Road

In the game *Settlers of Catan*, several players compete to gain the most points by building items on a hexagonal grid. You and two friends are playing on the grid shown in Figure C.1. The board consists of seven regular hexagons which define 24 *intersections* (vertices of the hexagons) and 30 *lanes* (edges of the hexagons) connecting them.

In the game, each player can build road segments on the lanes. Each road segment is owned by exactly one player. At most one road segment can be built on each lane. A player has a road of length k if there is a sequence of $k + 1$ (not necessarily distinct) intersections I_0, I_1, \dots, I_k such that the player owns k distinct road segments that connect intersections I_j and I_{j+1} (for each $j \in \{0, 1, \dots, k-1\}$).

There is a reward for the player who has the longest road. What is the length of each player's longest road?



Figure C.1: The game board.

Input

The input file contains several test cases, each of them as described below.

The input consists of a text representation of the game board. The game board consists of 19 lines and has size 19×31 . Each intersection is represented by one of ' $>$ ' or ' $<$ '. If a lane does not have a road segment on it, then all characters on that lane are one of '-', '/', or '\'. If a lane does have a road segment on it, then all characters on that lane are one of '1', '2' or '3', representing the player that owns the road segment on that lane. All other characters of the input are '.'.

The game board with no road segments is shown in the first sample input. It is guaranteed that any differences between that empty game board and the input are full lanes replaced with one of '1', '2' or '3'.

Output

For each test case, on a line by itself, display the length of the longest road owned by each of the three players (Player 1 first, then Player 2, then Player 3).

Sample Input

```

.....>-----<.....
...../.....\.....
...../.....\.....
...>-----<.....>-----<...
../.....\...../.....\.....
../.....\...../.....\.....
<.....>-----<.....>
.\...../.....\...../.....
.\...../.....\...../.....
...>-----<.....>-----<...

```

```

../. . . . . \. . . . . /. . . . . \. .
./ . . . . . \. . . . . /. . . . . \.
<. . . . . >-----<. . . . . >
.\. . . . . /. . . . . \. . . . . /.
.. \. . . . . /. . . . . \. . . . . /.
... >-----<. . . . . >-----<. .
. . . . . \. . . . . /. . . . .
. . . . . \. . . . . /. . . . .
. . . . . >-----<. . . . .
. . . . . >-----<. . . . .
. . . . . /. . . . . \. . . . .
. . . . . /. . . . . \. . . . .
... >11111<. . . . . >-----<. .
./ . . . . 1. . . . . /. . . . . \.
./ . . . . 1. . . . . /. . . . . \.
<. . . . . >11111<. . . . . >
.\. . . . 2. . . . . \. . . . . /.
.. \. . . . 2. . . . . \. . . . . /.
... >22222<. . . . . >-----<. .
./ . . . . \. . . . . /. . . . . \.
./ . . . . \. . . . . /. . . . . \.
<. . . . . >-----<. . . . . >
.\. . . . . /. . . . . \. . . . . /.
.. \. . . . . /. . . . . \. . . . . /.
... >-----<. . . . . >-----<. .
. . . . . \. . . . . /. . . . .
. . . . . \. . . . . /. . . . .
. . . . . >11111<.
. . . . . >-----<.
. . . . . /. . . . . 2.
. . . . . /. . . . . 2.
... >-----<. . . . . >-----<. .
./ . . . . \. . . . . /. . . . . \.
./ . . . . \. . . . . /. . . . . \.
<. . . . . >-----<. . . . . >
.\. . . . . /. . . . . \. . . . . /.
.. \. . . . . /. . . . . \. . . . . /.
... >-----<. . . . . >-----<. .
./ . . . . \. . . . . /. . . . . \.
./ . . . . \. . . . . /. . . . . \.
<. . . . . >-----<. . . . . >
.\. . . . 1. . . . . \. . . . . /.
.. \. . . . 1. . . . . \. . . . . /.
... >-----<. . . . . >-----<. .
. . . . . \. . . . . 3.
. . . . . \. . . . . 3.
. . . . . >-----<.
. . . . . >11111<.
. . . . . 1. . . . . 1.
. . . . . 1. . . . . 1.

```

```
...>11111<.....>11111<...  
..1.....1.....1.....1..  
.1.....1.....1.....1.  
<.....>11111<.....>  
.1.....1.....1.....1.  
..1.....1.....1.....1..  
...>11111<.....>11111<...  
..1.....1.....1.....1..  
.1.....1.....1.....1.  
<.....>11111<.....>  
.1.....1.....1.....1.  
..1.....1.....1.....1..  
...>11111<.....>11111<...  
.....1.....1.....  
.....1.....1.....  
.....>11111<.....
```

Sample Output

```
0 0 0  
3 2 0  
1 1 1  
25 0 0
```