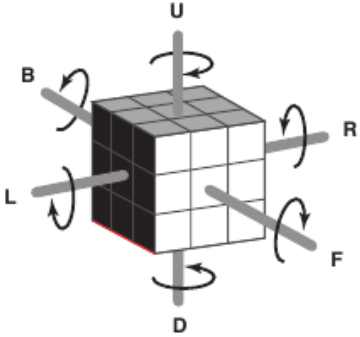


12492 Rubik Cycle

Probably everyone in this competition knows the Rubik's Cube, a challenging 3-D puzzle. Each of its six sides is covered with nine labels, each label of a color (blue, yellow, orange, white, green and red). In its initial state, all nine labels on one face have the same color. One ingenious mechanism enables each face to be independently rotated, causing the colors of the labels on the sides to be mixed.

Each of the faces of the Rubik's Cube is denoted by a letter: F, B, U, D, L, and R, as illustrated in the figure on the right.



The rotation of a face is called a *movement*. We describe the movements using the letters that identify the faces:

- a capital letter represents a 90° clockwise turn of the corresponding face;
- a lowercase letter represents a 90° counterclockwise turn of the corresponding face.

For example, F represents a 90° clockwise turn of face F; r represents a 90° counterclockwise turn of face R. A sequence of movements is denoted by a sequence of letters identifying faces. Thus, rDF represents a 90° counterclockwise turn of face R, followed by a 90° clockwise turn of face D, followed by a 90° clockwise turn of face F.

An interesting property of the Rubik's Cube is that any sequence of movements, if applied repeatedly, causes the cube to return to its original state (the state it had before the first application of the sequence). For example, after four applications of the sequence B the cube returns to its original state.

You must write a program that, given a sequence of movements, determines the minimum number of complete applications of that sequence to make the cube return to its original state.

Input

The input contains several test cases. Each test case is described in a single line, which contains the sequence of movements.

Output

For each test case your program must print a single line, containing a single integer, indicating the minimum number of complete applications of the given sequence to make the cube return to its original state.

Restrictions

- Each sequence has at least one and at most 80 characters.

Sample Input

```
Rr
LLL
dl
RUUdBd
```

Sample Output

1
4
105
1260