

# The Game

John has been given a game as a birthday gift. The game takes place on a 3x3 board and uses 8 pawns, numbered 1-8. There are two stages in the game:

- setting the initial position;
- proper game.

Setting the initial position is made by putting the pawns on the board in a random sequence (one square should remain empty). The proper game consists of moves which consists of moving a pawn that is a neighbour (horizontal or vertical) of an empty square to that square. The goal of the game is to verify if the following winning position can be reached (0 denotes the empty square):

```
876
543
210
```

Write program that, given an initial position, computes the minimal number of moves that should be made to achieve the above winning position.

## Input

The first line contains the number of initial positions. Next lines contain these positions. Every initial position is given as follows:

```
abc
def
ghi
```

where a-i are the numbers of pawns or 0 (empty field).

## Output

For every initial position given in the input, write to the output the minimal number of moves that should be made to reach the winning position. If the winning position is not attainable then write -1.

## Example 1

**Input:**

```
2
876
543
210
876
543
120
```

**Output:**

```
0
-1
```

## Example 2

**Input:**

2  
876  
543  
201  
876  
543  
102

**Output:**

1  
-1

## Points

The score awarded to your program is proportional to the number of correctly solved test cases (100 points is equivalent to the situation in which all tests have been solved correctly).

The number of points given in the ranking is scaled so that it is equal to 10 for the registered contestant whose solution has the highest score, and proportionally less for all solutions with lower scores.