#codefor15

Challenge 8.1: Glitch In The Matrix

You've got a **5 × 5** matrix, consisting of **24** zeroes and a **single** number one. Let's index the matrix rows by numbers from 1 to 5 from **top to bottom**, let's index the matrix columns by numbers from 1 to 5 from **left to right**. In one move, you are allowed to apply **one** of the two following transformations to the matrix :

- Swap two neighboring matrix rows, that is, rows with indexes **i** and **i + 1** for some integer **i (1 ≤ i < 5).**
- Swap two neighboring matrix columns, that is, columns with indexes **j** and **j + 1** for some integer **j (1 ≤ j < 5).**

If one is not located in the middle of the matrix we call it **"Glitch in the matrix "**. Your task is to make the matrix **glitch free** and Count the minimum number of moves needed to make the matrix glitch free.

Input Constraints:

The input consists of five lines, each line contains five integers: the **j-th** integer in the **i-th** line of the input represents the element of the matrix that is located on the intersection of the **i-th** row and the **j-th** column.

It is guaranteed that the matrix consists of 24 zeroes and a single number one.

Output Constraint:

Print a single integer : the minimum number of moves needed to remove the glitch in the matrix (to make the matrix glitch free).

Examples:

Input 1:

```
0 0 0 0 0
0 0 0 0 1
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
```

Output 1:

3

Input 2:

```
0 0 0 0 0
0 0 0 0 0
0 1 0 0 0
0 0 0 0 0
0 0 0 0 0
```

Output 2:

1