Forming a Magic Square ★

We define a magic square to be an $n \times n$ matrix of distinct positive integers from 1 to n^2 where the sum of any row, column, or diagonal of length n is always equal to the same number: the magic constant.

You will be given a 3×3 matrix s of integers in the inclusive range $[1, 9]$. We can convert any digit a to any other digit b in the range $[1, 9]$ at cost of $|a - b|$. Given s, convert it into a magic square at minimal cost. Print this cost on a new line.

Note: The resulting magic square must contain distinct integers in the inclusive range $[1, 9]$.

Example

$s = [[5, 3, 4], [1, 5, 8], [6, 4, 2]]

The matrix looks like this:

```
5 3 4
1 5 8
6 4 2
```

We can convert it to the following magic square:

```
8 3 4
1 5 9
6 7 2
```

This took three replacements at a cost of $|5 - 8| + |8 - 9| + |4 - 7| = 7$.

Function Description

Complete the formingMagicSquare function in the editor below.

formingMagicSquare has the following parameter(s):

- int s[3][3]: a 3×3 array of integers

Returns

- int: the minimal total cost of converting the input square to a magic square

Input Format

Each of the 3 lines contains three space-separated integers of row $s[i]$.

Constraints

- $s[i][j] \in [1, 9]$

Sample Input 0

```
4 9 2
3 5 7
8 1 5
```

Sample Output 0

```
1
```

Explanation 0

If we change the bottom right value, $s[2][2]$, from 5 to 6 at a cost of $|6-5|=1$, s becomes a magic square at the minimum possible cost.

Sample Input 1

```
4 8 2
4 5 7
6 1 6
```

Sample Output 1

```
4
```

Explanation 1

Using 0-based indexing, if we make

- $s[0][1]$->9 at a cost of $|9-8|=1$
- $s[1][0]$->3 at a cost of $|3-4|=1$
- $s[2][0]$->8 at a cost of $|8-6|=2$,

then the total cost will be $1+1+2=4$.

Change Theme Language Java 8

```
21
22      public static int formingMagicSquare(List<List<Integer>> s) {
23      // Write your code here
24          int a[][] = {
25              {8,1,6,3,5,7,4,9,2},
26              {6,1,8,7,5,3,2,9,4},
27              {4,9,2,3,5,7,8,1,6},
28              {2,9,4,7,5,3,6,1,8},
29              {8,3,4,1,5,9,6,7,2},
30              {4,3,8,9,5,1,2,7,6},
31              {6,7,2,1,5,9,8,3,4},
32              {2,7,6,9,5,1,4,3,8}
33          };
34
35          int ans = Integer.MAX_VALUE;
36
37          for(int i=0; i<8; i++){
38              int temp = 0;
39              for(int j=0; j<9; j++){
40                  temp += Math.abs(a[i][j] - s.get(j/3).get(j%3));
41              }
42
43              ans = Math.min(temp, ans);
44          }
45
46          return ans;
47      }
48
```

```
49  }
50
51  public class Solution {
```

Line: 79 Col: 1

Upload Code as File Test against custom input Run Code Submit Code

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

- Test case 0
- Test case 1
- Test case 2
- Test case 3
- Test case 4
- Test case 5
- Test case 6

Compiler Message

Success

Input (stdin) Download

```
4 9 2
3 5 7
8 1 5
```

Expected Output Download

```
1
```