





Forming a Magic Square ★

229 more points to get your next star!
Rank: 628696 | Points: 246/475

Problem Submissions Leaderboard Editorial 🖰

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 $\mathbf{3} \times \mathbf{3}$ array of integers

Returns

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

Input Format

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

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[0][0]-93 at a cost of |3-4|=1 *s[1][0]-98 at a cost of |8-6|=2, then the total cost will be 1+1+2=4.
```

```
₩ K Z
                                          Change Theme Language Java 8
     import java.io.*;
 1
 2
     import java.util.*;
 3
 4
     public class Solution {
 5
 6
         public static void main(String[] args) {
 7
             /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your
     class should be named Solution. */
             Scanner scan = new Scanner(System.in);
 8
             int[] square = new int[9];
 9
             for (int i = 0; i < 9; i++) {
10
                  square[i] = scan.nextInt();
11
12
             int[][] matrix={{4,9,2,3,5,7,8,1,6},
13
14
                              \{2,7,6,9,5,1,4,3,8\},
                              \{6,1,8,7,5,3,2,9,4\},
15
                              \{8,3,4,1,5,9,6,7,2\},
16
17
                              {2,9,4,7,5,3,6,1,8},
                              \{6,7,2,1,5,9,8,3,4\},
18
                              \{8,1,6,3,5,7,4,9,2\},
19
20
                              {4,3,8,9,5,1,2,7,6};
21
22
             int minOff = 99;
             for (int i = 0; i < 8; i++) {
23
                                                                                       Line: 35 Col: 2
```

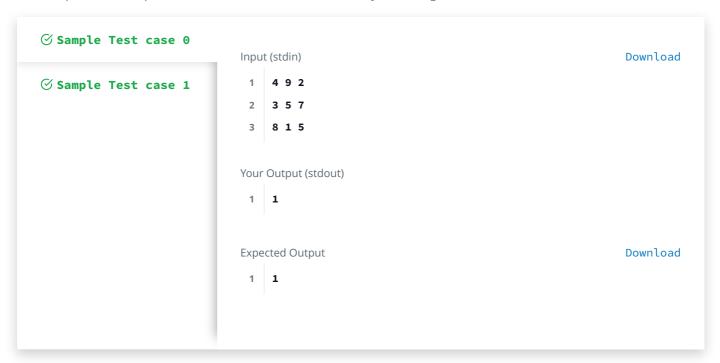
Test against custom input

Run Code

Submit Code

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.



Contest Calendar | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature