Forming a Magic Square

by pkacprzak

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 (i.e., the *magic constant*).

Consider 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 $|a - b|$.

Given s, convert it into a magic square at *minimal* cost by changing zero or more of its digits. Then print this cost on a new line.

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

Input Format

There are 3 lines of input. Each line describes a row of the matrix in the form of 3 space-separated integers denoting the respective first, second, and third elements of that row.

Constraints

- All integers in s are in the inclusive range $[1, 9]$.

Output Format

Print an integer denoting the minimum cost of turning matrix s into a magic square.

Sample Input

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

Sample Output

```
1
```

Explanation

Matrix s initially looks like this:

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

Observe that it's not yet magic, because not all rows, columns, and center diagonals sum to the same number.

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. Thus, we print the cost, 1, on a new line.

Submissions: 980
Max Score: 20
Difficulty: Easy

Rate This Challenge:

More

Current Buffer (saved locally, editable) C#

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
class Solution {

    static void Main(String[] args) {

            List<string> entrada = new List<string>();
            for (int i = 0; i < 3; i++)
            {
                entrada.Add(String.Join("", Console.ReadLine().Split(' ')));
            }
            //foreach (string s in entrada)
            //{
            // Console.WriteLine(s);
            //}
            List<List<string>> todos = new List<List<string>>();
            todos.Add(new List<string>(new string[] { "816", "357", "492" }));
            todos.Add(new List<string>(new string[] { "618", "753", "294" }));
            todos.Add(new List<string>(new string[] { "438", "951", "276" }));
            todos.Add(new List<string>(new string[] { "276", "951", "438" }));
            todos.Add(new List<string>(new string[] { "294", "753", "618" }));
            todos.Add(new List<string>(new string[] { "492", "357", "816" }));
            todos.Add(new List<string>(new string[] { "672", "159", "834" }));
            todos.Add(new List<string>(new string[] { "834", "159", "672" }));
            int min_costo = int.MaxValue;
            foreach (List<string> lista in todos)
            {
                int costo = 0;
                for (int i = 0; i < 3; i++)
                {
                    for (int j = 0; j < 3; j++)
                    {
                        costo += Math.Abs(int.Parse(entrada[i][j].ToString()) - int.Parse(lista[i]
                        [j].ToString()));
                    }
                }
                min_costo = Math.Min(min_costo, costo);
            }
            Console.WriteLine(min_costo);
    }
}

```

Line: 41 Col: 42

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

Congrats, you solved this challenge!

- ✔ Test Case #0
- ✔ Test Case #1
- ✔ Test Case #2
- ✔ Test Case #3
- ✔ Test Case #4
- ✔ Test Case #5

- ✔ Test Case #6
- ✔ Test Case #7
- ✔ Test Case #8
- ✔ Test Case #9
- ✔ Test Case #10
- ✔ Test Case #11
- ✔ Test Case #12
- ✔ Test Case #13
- ✔ Test Case #14
- ✔ Test Case #15
- ✔ Test Case #16
- ✔ Test Case #17
- ✔ Test Case #18
- ✔ Test Case #19
- ✔ Test Case #20
- ✔ Test Case #21

Next Challenge