Magic Square Forming

locked

by pkacprzak

Problem Submissions Discussions Editorial

Consider a 3×3 matrix, s, of integers in the inclusive range $[1, 9]$. Any digit, a, can be changed to any other digit, b, in the range $[1, 9]$ at cost $|a - b|$.

Given matrix s, convert it into a magic square by changing zero, one, or more of the digits in s. You must do this in such a way that the cost is minimal and then print the minimum possible 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 a single integer denoting the smallest possible 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 will become a magic square at the minimum possible cost. Thus, we print the cost, 1, on a new line.

Submissions: 227
Max Score: 100

Rate This Challenge:

More

Current Buffer (saved locally, editable) C#

```
using System;
```

```
using System;
using System.Collections.Generic;
using System.IO;
class Solution {
    static void Main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution */
         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: 36 Col: 25

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

Testcase 0 ✔

Congratulations, you passed the sample test case.

Click the **Submit Code** button to run you code against all the test cases.

Input (stdin)

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

Your Output (stdout)

```
1
```

Expected Output

```
1
```