**MAGIC SQUARE COST**

#include<stdio.h>

int main(void)

{

    int i = 0;

    int j = 0;

    int magic\_mat[8][3][3] =

    {

    {{8, 1, 6}, {3, 5, 7}, {4, 9, 2}},

    {{6, 1, 8}, {7, 5, 3}, {2, 9, 4}},

    {{4, 9, 2}, {3, 5, 7}, {8, 1, 6}},

    {{2, 9, 4}, {7, 5, 3}, {6, 1, 8}},

    {{8, 3, 4}, {1, 5, 9}, {6, 7, 2}},

    {{4, 3, 8}, {9, 5, 1}, {2, 7, 6}},

    {{6, 7, 2}, {1, 5, 9}, {8, 3, 4}},

    {{2, 7, 6}, {9, 5, 1}, {4, 3, 8}},

    };

int A[3][3];

for (int i = 0; i < 3; i++)

{

    for (int j = 0; j < 3; j++)

    {

        scanf("%d", &A[i][j]);

    }

}

int min\_cost = 81;

for (int k = 0; k < 8; k++)

{

    int crt\_cost = 0;

    for (int i = 0; i < 3; i++)

    {

        for (int j = 0; j < 3; j++)

        {

            crt\_cost += abs( A[i][j] - magic\_mat[k][i][j] );

        }

    }

    if (crt\_cost < min\_cost)

    {

        min\_cost = crt\_cost;

    }

}

printf("%d", min\_cost);

    return 0 ;

}

**ANS:**

**Sample Input**

4 9 2

3 5 7

8 1 5

**Sample Output**

1

**Explanation**

**If we change the bottom right value s[2][2],  from5  to 6 at a cost of |6-5|=1 , s  becomes a magic square at the minimum possible cost.**