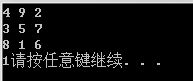
**9-1九宫格**

问题描述

　　九宫格。输入1-9这9个数字的一种任意排序，构成3\*3二维数组。如果每行、每列以及对角线之和都相等，打印1。否则打印0。

样例输出

与上面的样例输入对应的输出。  
例：  


数据规模和约定

　　输入1-9这9个数字的一种任意排序。

C++代码：

#include<iostream>

#include<string.h>

using namespace std;

main()

{

int all[3][3];

for(int i=0;i<3;i++)

for(int j=0;j<3;j++)

scanf("%d",&all[i][j]);

int val[8];

memset(val,0,sizeof(val));

for(int i=0;i<3;i++)

for(int j=0;j<3;j++)

val[i]+=all[i][j];

for(int i=3;i<6;i++)

for(int j=0;j<3;j++)

val[i]+=all[j][i-3];

val[6]=all[0][0]+all[1][1]+all[2][2];

val[7]=all[0][2]+all[1][1]+all[2][0];

int flag=1;

for(int i=1;i<8;i++)

if(val[0]!=val[i])

flag=0;

if(flag==0)

printf("0\n");

else

printf("1\n");

}

C代码：

#include<stdio.h>

#include<string.h>

int main()

{

int e,q,s,t;

int a[3][3];

for(e=0;e<3;e++)

for(q=0;q<3;q++)

scanf("%d",&a[e][q]);

t=a[0][0]+a[0][1]+a[0][2];

for(e=0;e<3;e++)

{

s=0;

for(q=0;q<3;q++)

{

s=a[e][q]+s;

}

if(s!=t)

{

printf("0");

return 0;

}

}

for(e=0;e<3;e++)

{

s=0;

for(q=0;q<3;q++)

{

s=a[q][e]+s;

}

if(s!=t)

{

printf("0");

return 0;

}

}

if(t!=a[0][0]+a[1][1]+a[2][2]||t!=a[0][2]+a[1][1]+a[2][0])

{

printf("0");

return 0;

}

printf("1\n");

return 0;

}

JAVA代码：

import java.util.Scanner;

public class Main {

public static void main(String arg[]) {

Scanner sn = new Scanner(System.in);

int num[] = new int[9];

int count[] = new int[8];

for (int i = 0; i < 9; i++) {

num[i] = sn.nextInt();

}

for (int i = 0; i < 3; i++) {

count[i] = num[3 \* i + 0] + num[3 \* i + 1] + num[3 \* i + 2];

}

for (int i = 0; i < 3; i++) {

count[i + 3] = num[i] + num[i + 3] + num[i + 6];

}

for (int i = 0; i < 3; i++) {

count[6] = count[6] + num[i \* 3 + i];

count[7] = count[7] + num[i \* 3 + 2 - i];

}

for (int i = 0; i < 7; i++) {

if (count[i] != count[i + 1]) {

System.out.println(0);

break;

}

if (i == 6) {

System.out.println(1);

}

}

}

}