

## 牛客网-华为机试练习题 84

### 题目描述

有一个数据表格为二维数组（数组元素为int类型），行长度为ROW\_LENGTH,列长度为COLUMN\_LENGTH。对该表格中数据的操作可以在单个单元内，也可以对一个整行或整列进行操作，操作包括交换两个单元中的数据；插入某些行或列。

请编写程序，实现对表格的各种操作，并跟踪表格中数据在进行各种操作时，初始数据在表格中位置的变化轨迹。

详细要求:

- 1.数据表规格的表示方式为“行\*列”，数据表元素的位置表示方式为[行,列]，行列均从0开始编号
- 2.数据表的最大规格为9行\*9列，对表格进行操作时遇到超出规格应该返回错误
- 3.插入操作时，对m\*n表格，插入行号只允许0~m，插入列号只允许0~n。超出范围应该返回错误
- 4.只需记录初始表格中数据的变化轨迹，查询超出初始表格的数据应返回错误

例如: 初始表格为4\*4，可查询的元素范围为[0,0]~[3,3]，假设插入了第2行，数组变为5\*4，查询元素[4,0]时应该返回错误

- 5.查询数据要求返回一个链表，链表中节点的顺序即为该查询的数据在表格中的位置变化顺序（需包含初始位置）

### 输入描述:

输入数据按下列顺序输入：

- 1 表格的行列值
- 2 要交换的两个单元格的行列值
- 3 输入要插入的行的数值
- 4 输入要插入的列的数值
- 5 输入要获取运动轨迹的单元格的值

### 输出描述:

输出按下列顺序输出：

- 1 初始化表格是否成功，若成功则返回0， 否则返回-1
- 2 输出交换单元格是否成功
- 3 输出插入行是否成功
- 4 输出插入列是否成功
- 5 输出要查询的运动轨迹的单元查询是否成功

示例1

输入

```
3 4
1 1
0 1
2
1
2 2
```

输出

```
0
```

0  
0  
0  
0

## 解决代码:

```
import java.util.Scanner;

/**
 * Created by dy on 2016/7/6.
 */
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        while (scanner.hasNext()) {
            int[] arr = new int[10];
            for (int i = 0; i < 10; i++) {
                arr[i] = scanner.nextInt();
            }
            System.out.print(solve(arr));
        }
        scanner.close();
    }

    private static String solve(int[] arr) {
        int[] result = new int[5];
        // 检查行列值
        if (arr[0] < 0 || arr[0] > 9 || arr[1] < 0 || arr[1] > 9) {
            result[0] = -1;
        } else {
            result[0] = 0;
        }
        // 检查交换单元格是否合法
        if (result[0] == 0 && (arr[2] >= 0 && arr[2] < arr[0] && arr[3] >= 0 && arr[3] <
arr[1])) {
            && (arr[4] >= 0 && arr[4] < arr[0] && arr[5] >= 0 && arr[5] < arr[1])) {
                result[1] = 0;
            } else {
                result[1] = -1;
            }
        }
        // 检查插入行是否成功
        if (result[0] == 0 && (arr[6] >= 0 && arr[6] < arr[0])) {
            result[2] = 0;
        } else {
            result[2] = -1;
        }
        // 检查插入列是否成功
        if (result[0] == 0 && (arr[7] >= 0 && arr[7] < arr[1])) {
            result[3] = 0;
        } else {
            result[3] = -1;
        }
        // 检查访问是否成功
        if (result[0] == 0 && (arr[8] >= 0 && arr[8] < arr[0] && arr[9] >= 0 && arr[9] <
arr[1])) {
            result[4] = 0;
        } else {
            result[4] = -1;
        }
    }
}
```

```
    StringBuilder b = new StringBuilder();  
    for (int i : result) {  
        b.append(i).append('\n');  
    }  
    return b.toString();  
}  
}
```