Diagonal Difference ☆

9 more points to get your first star!

Rank: **1656636** | Points: **21/30**

Problem Solving

Problem Submissions Leaderboard Editorial

Given a square matrix, calculate the absolute difference between the sums of its diagonals.

For example, the square matrix arr is shown below:

```
1 2 3
4 5 6
9 8 9
```

The left-to-right diagonal = $1 + 5 + 9 = 15$. The right to left diagonal = $3 + 5 + 9 = 17$. Their absolute difference is $|15 - 17| = 2$.

Function description

Complete the *diagonalDifference* function in the editor below. It must return an integer representing the absolute diagonal difference.

diagonalDifference takes the following parameter:

- arr: an array of integers .

Input Format

The first line contains a single integer, n, the number of rows and columns in the matrix arr.
Each of the next n lines describes a row, $arr[i]$, and consists of n space-separated integers $arr[i][j]$.

Constraints

- $-100 \leq arr[i][j] \leq 100$

Output Format

Print the absolute difference between the sums of the matrix's two diagonals as a single integer.

Sample Input

```
3
11 2 4
4 5 6
10 8 -12
```

Sample Output

```
15
```

Explanation

The primary diagonal is:

```
11
   5
     -12
```

Sum across the primary diagonal: 11 + 5 - 12 = 4

The secondary diagonal is:

```
     4
   5
10
```

Sum across the secondary diagonal: 4 + 5 + 10 = 19

Difference: |4 - 19| = 15

Note: |x| is the absolute value of x

Change Theme Java 8

```
18       * The function is expected to return an INTEGER.
19       * The function accepts 2D_INTEGER_ARRAY arr as parameter.
20       */
21
22      public static int diagonalDifference(List<List<Integer>> arr) {
23      // Write your code here
24          int rows = arr.size();
25          int diagPrincipal = 0, diagSec = 0;
26          int result;
27          for(int i = 0; i<rows; i++) {
28              for(int j = 0; j<rows; j++) {
29                  if(i==j)
30                      diagPrincipal += arr.get(i).get(j) ;
31
32                  if(i+j == rows -1)
33                      diagSec+= arr.get(i).get(j) ;
34              }
35          }
36          result = diagPrincipal - diagSec;
37          return result < 0 ? result * -1 : result;
38      }
39
40  }
41
42  public class Solution {
43      public static void main(String[] args) throws IOException {
44          BufferedReader bufferedReader = new BufferedReader(new InputStreamReader
    (System.in));
45          BufferedWriter bufferedWriter = new BufferedWriter(new FileWriter(System.getenv
    ("OUTPUT_PATH")));
46
47          int n = Integer.parseInt(bufferedReader.readLine().trim());
48
49          List<List<Integer>> arr = new ArrayList<>();
```

Line: 37 Col: 50

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

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Sample Test case 0

Input (stdin) Download

```
1  3
2  11 2 4
```

```
3 4 5 6
4 10 8 -12
```

Your Output (stdout)

```
1 15
```

Expected Output

```
1 15
```

Download