

Amechi Aduba
9/6/23
Clark University
CS 160

Diagonal Difference

```
9  class Result {
10
11  /*
12   * Complete the 'diagonalDifference' function below.
13   *
14   * The function is expected to return an INTEGER.
15   * The function accepts 2D_INTEGER_ARRAY arr as parameter.
16   */
17
18  public static int diagonalDifference(List<List<Integer>> arr) {
19  // Write your code here
20      int first_diagonal = 0;
21      int second_diagonal = 0;
22
23      for (int i = 0; i < arr.size(); i++){
24          first_diagonal += arr.get(i).get(i);
25      }
26      for (int i = 0; i < arr.size(); i++){
27          second_diagonal += arr.get(arr.size() - i - 1).get(i);
28      }
29
30      int difference = first_diagonal - second_diagonal;
31      return Math.abs(difference);
32  }
```

✓ Test case 0

✓ Test case 1

✓ Test case 2

✓ Test case 3

✓ Test case 4

✓ Test case 5

✓ Test case 6

Compiler Message

Success

Input (stdin) [Download](#)

1	3
2	11 2 4
3	4 5 6
4	10 8 -12

Expected Output [Download](#)

1	15
---	----

Staircase

```
6  import java.util.concurrent.*;
7  import java.util.regex.*;
8
9  class Result {
10
11      /*
12       * Complete the 'staircase' function below.
13       *
14       * The function accepts INTEGER n as parameter.
15       */
16
17      public static void staircase(int n) {
18          // Write your code here
19          for (int i = 1; i < n + 1; i++){
20              for (int j = 0; j < n - i; j++){
21                  System.out.print(" ");
22              }
23              for (int z = 0; z < i; z++){
24                  System.out.print("#");
25              }
26              System.out.println("");
27          }
28      }
29  }
```

Fetching Results

✓ Test case 0  ✓ Test case 4  ✓ Test case 8 

✓ Test case 1  ✓ Test case 5 

✓ Test case 2  ✓ Test case 6 

✓ Test case 3  ✓ Test case 7 