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.

diagonalDifference takes the following parameter:

• int arr[n][m]: an array of integers

Return

• int: the absolute diagonal difference

Input Format

The first line contains a single integer, n, the number of rows and columns in the square matrix arr.

Each of the next n lines describes a row, arr[i], and consists of n space-separated integers arr[i][j].

Constraints

•
$$-100 \le arr[i][j] \le 100$$

Output Format

Return 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:

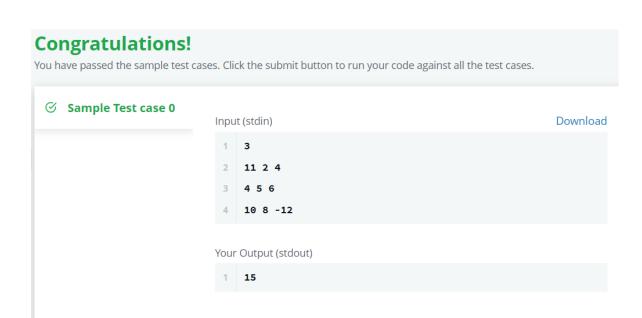
```
4
5
10
```

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

Difference: |4 - 19| = 15

Note: |x| is the absolute value of x

```
#!/bin/python3
import math
import os
import random
import re
import sys
# Complete the 'diagonalDifference' function below.
# The function is expected to return an INTEGER.
# The function accepts 2D INTEGER ARRAY arr as parameter.
def diagonalDifference(arr):
    # Write your code here
    s=0
    s1 = 0
    x=0
    while x<len(arr):</pre>
        s=s+arr[x][x]
        s1=s1+arr[x][len(arr)-x-1]
        x=x+1
    return abs(s-s1)
if name == ' main ':
    fptr = open(os.environ['OUTPUT PATH'], 'w')
   n = int(input().strip())
    arr = []
    for in range(n):
        arr.append(list(map(int, input().rstrip().split())))
    result = diagonalDifference(arr)
    fptr.write(str(result) + '\n')
    fptr.close()
```



Download

Expected Output

